



LISTADO GENÉRICO DE SAGA

Manual de USO

www.saga-consulting.net

info.saga@saga-consulting.net

C/ Salado, 11. Local 10. 41011 Sevilla

+34 954 45 72 75

CONTROL DEL DOCUMENTOS

Realizado

(16/05/2008)

Noel
Guillén
Montilla

Revisado

(11/09/2008)

Sergio
Raposo
Vargas

Director Técnico

Control de Modificaciones

<i>Rev.</i>	<i>Fecha</i>	<i>Autor/es</i>	<i>Descripción</i>
Nº 0	16/05/2008	Noel Guillen Montilla	Redacción inicial del documento
Nº 1	11/09/2008	Sergio Raposo Vargas	Corrección del documento

1	OBJETIVO	4
2	DESCRIPCIÓN Y CONFIGURACIÓN	5
3	GENERACIÓN DEL ELEMENTS	6
4	GENERACIÓN DEL RECURSO SGLISTADO	7
5	GENERACIÓN DE LA PLANTILLA HTML.....	10
5.1	BODY. _____	10
5.2	LISTADO GENERAL _____	11
5.3	LISTADO SIN FOTO _____	11
5.4	LISTADO IMPAR _____	11
5.5	LISTADO IMPAR SIN FOTO _____	12
5.6	PAGINACIÓN _____	12
5.7	SEPARADOR _____	13
6	CATEGORIZACIÓN.....	14
7	IMPLEMENTACIÓN DE UNA CATEGORIZACIÓN.....	15
8	ERRORES COMUNES.....	16
8.1	JAVA.LANG.STRINGINDEXOUTOFBOUNDSEXCEPTION: STRING INDEX OUT OF RANGE: -1 __	16
9	CONCLUSIONES.....	17



1 OBJETIVO

El objetivo de este manual es explicar la configuración y el uso del módulo de listado genérico desarrollado por Saga Consulting para las distintas versiones de OpenCms 7.

Este manual va destinado para los siguientes roles:

- Usuario editor: Se encargará de editar el recurso de tipo listado para gestionar las distintas secciones de la web que sean un listado.
- Programador: Se encargará de programar el código html que será finalmente mostrado en la web.
- Administrador: Los usuarios con roles de administración deberán encargarse de instalar y de configurar el módulo inicialmente.

Este manual ofrece información para todos y cada uno de estos tres tipos de usuarios.



2 DESCRIPCIÓN Y CONFIGURACIÓN

El componente genera un listado a partir de una ruta que contiene una lista de recursos y le aplica una plantilla HTML

Las plantillas HTML se encuentran agrupadas en una carpeta que se puede configurar en el archivo:

```
/system/modules/com.saga.opencms.templatesaga.module.listado/schemas/listado.xsd
```

Para ello se modifica la siguiente línea:

```
<layout element="Formato"  
widget="com.saga.opencms.templatesaga.widget.SelectUrlWidget"  
configuration="RUTA DE LAS PLANTILLAS HTML />
```

3 GENERACIÓN DEL ELEMENTS

Primero debemos importar el modulo “com.saga.opencms.templatesaga.module.listado.*”;

Posteriormente generamos un objeto tipo “SagaListado”.

El objeto recibirá la pagina actual, la ruta del archivo Sglistado, el objeto request, response y pagecontext.

El constructor quedaría de la siguiente manera:

```
SagaListado sagaListado= new SagaListado(paginaActual,ruta,request,response,pageContext);
```

Existe otro constructor similar al anterior que se utiliza para la categorización. Se construye como el anterior, pero además recibe un parámetro entero que indica la forma de mostrar el html para la categorización.

Para pintar el listado se ejecuta:

```
out.println(sagaListado.crearListado());
```

4 GENERACIÓN DEL RECURSO SGLISTADO

El recurso contara con los siguientes campos:

Title/Titulo (Opcional) : Titulo general para el listado.

Text/Contenido (Opcional): Descripción general del listado.

Footer/Pie (Opcional): Pie general del listado.

Image/Imagen (Opcional): Imagen general del listado.

Collector: Forma de recorrer el listado de recursos de acuerdo a la api de opencms.

Parámetros: Listado con los distintos campos a parsear. Consta de los siguientes parámetros:

- **Parámetro**: nombre del campo del recurso a mostrar. Dependerá del tipo de recurso. Por defecto tomara el campo del recurso con ese nombre, si se desease mostrar una propiedad, sería necesario preceder el valor de la misma con la etiqueta "property:".
- **Patrón**: texto por el que se sustituirá el parámetro en la plantilla html. Deberá existir en la plantilla html el valor patrón de la siguiente forma: {PATRON}.

Nota: Este campo **NO** lleva los {}

- **Tipo**: Indica el tipo del campo que se trata.
- **Atributos(Opcional)**: Solo son necesarios para los tipos Enlace e Imagen.

Atributos del tipo imagen: existen dos tipos de atributos, los atributos ATTR_ que son los propios de la etiqueta <img:> de html; y los atributos SCALE_ que son los propios del tag <cms:img> para escalar las imágenes.

Para asignar un valor variable a los atributos de tipo ATTR_, se añade el símbolo "=" precedido de el nombre de la variable, en caso contrario se puede escribir un valor constante sin especificarle ninguna marca de inicio.

Atributos del tipo enlace:

ENLACE_target: Indica donde se abrirá el enlace. "BLANK" para nueva ventana

Enlace _ descripción: Texto del enlace a mostrar por defecto.

ENLACE_campo: Campo del que se recogerá el texto a mostrar en el enlace. Si se rellena este atributo se ignorara en Enlace _ descripción.

Si ninguna de los atributos son rellenos, aparecerá como texto la uri del enlace y el target vacío.

Atributos del tipo texto:

TEXTO_MAX_CARACTERES: Indica el numero de caracteres que se mostraran, añadiendo puntos suspensivos en caso de que la oración continúe.

Resource/Ruta: Ruta donde se encuentran los recursos a listar.

ResourceType/Tipo: tipo del elemento al listar. Se marcara la opción todos para listar todos los recursos pertenecientes a una categoría, por lo que es necesario haber activado la casilla de Categorización previamente.

NumResources/Nº Elementos: Numero de recursos a listar.

PageSize/Tamaño pagina (OPCIONAL): Número de elementos por página.

PageNavLength/Nº paginas (OPCIONAL): Número de elementos en la navegación

Patrón: Patrón que seguirá para crear un nuevo recurso.

Formato: Plantilla HTML a utilizar.

Par-Impar: Indica si tendrá distintos estilos para pares e impares. (ver plantilla para más información).

Editable: indica si se podrá editar desde el HTML.

Categorización: indica si esta activa la categorización

Categoría: Categoría y subcategoría de elementos que serán listados.

Order: Permite ordenar una lista de recursos categorizados atendiendo a diferentes criterios

El formulario quedará de la siguiente forma:

Idioma: español

Título:

Contenido:

Pie:

Imagen:

Collector:

Parámetros:

Patrón:

Tipo:

Atributos:

Ruta:

Tipo:

Nº elementos:

Tamaño página:

Nº páginas:

Patrón:

Formato:

Par. Impar:

Editable:

5 GENERACIÓN DE LA PLANTILLA HTML

La plantilla estará formada por las siguientes partes, donde **Body y listado general deberán aparecer obligatoriamente**.

5.1 BODY.

Esta plantilla **es obligatoria**.

Se define de la siguiente manera:

```
<!--START_TEMPLATE_BODY-->  
<!--END_TEMPLATE_BODY-->
```

Consta de los siguientes elementos:

{TEMPLATE_TITLE}: Patrón donde se parseara la propiedad title.

Es opcional. Vacío por defecto

{TEMPLATE_CONTENIDO}: Patrón donde se parseara la propiedad text.

Es opcional. Vacío por defecto

{TEMPLATE_PAGINACION}: Patrón donde se parseara la paginación, si se encuentra activada, sino aparecerá vacío. Es opcional.

{TEMPLATE_FOOTER}: Patrón donde se parseara la propiedad footer.

Es opcional. Vacío por defecto

: Patrón donde se parseara el src de la imagen.

Es opcional. Vacío por defecto

{TEMPLATE_LISTADO}: Patrón donde se parseara el html del listado.

Es obligatorio, para que se pueda ver el listado

{TEMPLATE_CATEGORIAS}: Patrón donde se parseará el select de las categorías.

Es opcional. Útil solo si está la categorización activada.

5.2 LISTADO GENERAL

Es la plantilla que se tomará por defecto para listar los elementos. **Es obligatoria**

Se define de la siguiente manera:

```
<!--START_TEMPLATE_PAR-->  
<!--END_TEMPLATE_PAR-->
```

El patrón **{uri}** será **obligatorio** si se pretende que aparezca un link al detalle de la noticia

El resto de patrones serán los especificados en la lista de parámetros de sglistado marcados mediante {}.

Los tipos ENLACE e IMAGEN generarán el html completo. (No poner o <ref.>....)

5.3 LISTADO SIN FOTO

Esta plantilla será opcional y se tomara en caso de que un elemento no tenga foto.

Se defina de la siguiente manera:

```
<!--START_TEMPLATE_PAR_SIN-->  
<!--END_TEMPLATE_PAR_SIN-->
```

Los patrones son iguales a los de la plantilla general

5.4 LISTADO IMPAR

Esta plantilla será opcional y se tomara en caso de que se active la opción par/impar.

Si no se encuentra se tomara la plantilla general por defecto.

Se defina de la siguiente manera:

```
<!--START_TEMPLATE_IMPAR-->  
<!--END_TEMPLATE_IMPAR-->
```

Los patrones son iguales a los de la plantilla general

5.5 LISTADO IMPAR SIN FOTO

Esta plantilla será opcional y se tomara en caso de que se active la opción par/impar para los elementos que no tienen imagen.

Si no se encuentra se tomara la plantilla impar por defecto.

Se defina de la siguiente manera:

```
<!--START_TEMPLATE_IMPAR_SIN-->  
<!--END_TEMPLATE_IMPAR_SIN-->
```

Los patrones son iguales a los de la plantilla general

5.6 PAGINACIÓN

Esta plantilla es opcional y se tomara si se encuentra activada la paginación.

Si no se encuentra la plantilla se generara la paginación por defecto

Se defina de la siguiente manera:

```
<!--START_TEMPLATE_PAGINACION-->  
<!--END_TEMPLATE_PAGINACION-->
```

Los patrones que se le pueden aplicar son los siguientes:

{PRIMERA_PAGINA}: indicara la ruta hacia la primera página.

```
(<a href src="{PRIMERA_PAGINA}" ...)
```

{ANTERIOR_PAGINA}: indicara la ruta hacia la página anterior.

```
(<a href src="{ ANTERIOR_PAGINA }" ...)
```

{PA} {/PA}: El html contenido entre estas etiquetas se ocultara para la pagina 1. Obligatorio si aparecen los patrones {PRIMERA_PAGINA} o { ANTERIOR_PAGINA }

{PAGINACION_CENTRAL}: Mostrara el listado numérico de páginas.

{ULTIMA_PAGINA}: indicara la ruta hacia la última página.

```
(<a href src="{ULTIMA_PAGINA}" ...)
```

{SIGUIENTE_PAGINA}: indicara la ruta hacia la página siguiente.

```
(<a href src="{ SIGUIENTE_PAGINA }" ...)
```

{VA} {/VA}: El html contenido entre estas etiquetas se ocultara para la última página. Obligatorio si aparecen los patrones { ULTIMA _PAGINA } o { SIGUIENTE _PAGINA }

5.7 SEPARADOR

Se utilizará para separar el listado numérico de la {PAGINACION_CENTRAL}.

Por defecto se toma "-";

Se defina de la siguiente manera:

```
<!--START_TEMPLATE_SEPARADOR-->
```

```
,
```

```
<!--END_TEMPLATE_SEPARADOR-->
```

Donde "," sería el separador.

6 CATEGORIZACIÓN

Para mostrar una lista categorizada, los elementos a mostrar deben pertenecer a una categoría o subcategoría.

Las categorías se definirán en la ruta:

```
/system/categories/
```

Una categoría podrá contener todas las subcategorías que quiera, alcanzando el nivel de profundidad deseado. Los elementos podrán tener múltiples categorías

Para añadir un elemento a una categoría se podrá emplear dos procedimientos.

El primero consiste marcar con el botón derecho sobre el recurso y seleccionar Relations>AssignCategories. Posteriormente se seleccionarán todas las categorías deseadas.

El segundo método consiste en activar el campo category del recurso que es del tipo

```
com.saga.opencms.templatesaga.widget.CmsMultiCategoryWidget
```

El tipo del campo categoría en el recurso será OpenCmsString.

En este campo aparecerán todas las categorías existentes para que puedan ser seleccionadas.

Este campo podrá ser configurable desde el xsd del recurso añadiendo la ruta inicial de categorización en el apartado de configuración. Ejemplo:

```
<layout element="Category" widget="com.saga.opencms.templatesaga.  
widget.CmsMultiCategoryWidget" configuration="noticias/actualidad" />
```

En este caso actualidad y todas sus subcategorías aparecerían en el selec.

Nota: Al activar el campo category, se eliminaran todas las relaciones anteriores y solo se mantendrán las seleccionadas en el selec múltiple. NO se pueden utilizar los dos métodos indistintamente.

7 IMPLEMENTACIÓN DE UNA CATEGORIZACIÓN

Para implementar una nueva categorización, es necesario seguir los siguientes pasos:

- 1) Se define una constante en la clase SagaListado, indicando la forma de categorización que se va a usar con un valor determinado.
- 2) Se añade una entrada en el switch del método parsearElementosCategorias() con la clase encargada de generar el html de la categorización
- 3) Se construye la clase que debe implementar la interfaz IHtmlCategorización
- 4) Se implementa el método generaHTML, que recibe la categoría actual, la categoría base y un objeto CmsJspActionElement.

Para generar los enlaces, las rutas deben seguir la siguiente estructura:

```
"CmsJspTagLink.linkTagAction (cms.info ("opencms.request.uri")+?categoria="
```



8 ERRORES COMUNES

8.1 **JAVA.LANG.STRINGINDEXOUTOFBOUNDSEXCEPTION: STRING INDEX OUT OF RANGE: -1**

Si el error se produce al activar la categorización es posible que se deba a un mal paso de la ruta mediante parámetros a la hora de incluir los elements.

La ruta debe pasarse mediante código, de la siguiente forma:

```
"cms.getRequestContext().getUri()".
```

NO ADMITIENDO las variables de la siguiente forma:

```
%(opencms.uri)
```





9 CONCLUSIONES

El módulo de listado desarrollado por Saga nos ofrece un importante mecanismo de configuración de los portales desarrollados con OpenCms ya que permite, sin necesidad de tocar una sola línea de código, configurar un listado de contenidos de OpenCms, y poder aplicarle distinta forma de visualización gracias al uso de plantillas predefinidas.

En este manual se ha detallado el funcionamiento al completo del módulo, desde su configuración inicial hasta el significado de cada uno de los campos que conforman el recurso de Listado.

