

magnolia

Espacios

Personas

Preguntas

Calendarios

Crear

Buscar

Plantillas: FreeMarker

- Tipos de plantillas de introducción: páginas, áreas y componentes
- Plantilla de página == Definición de página
- Directivas de plantillas - cms.page
- 'Recursos' web estáticos y ayudante 'resfn'
- Diálogos: el modelo de contenido y el creador de contenido
- Acceso al contenido: el objeto 'contenido'
- Plantilla de área: configuración base
- Plantilla de componente == Definición de componente
- FreeMarker: conceptos básicos
- Funciones de plantillas
- Funciones de plantillas DAM
- Temas: Cambio de tamaño de imágenes para plantillas
- Vinculación en JCR y Magnolia
- Referencias de plantillas
- Paquetes de mensajes estáticos
- Contenido i18n
- Definición de área: uso de scripts de área
- Objeto 'Definición': Parámetros
 - (O) Aplicaciones de contenido: usar y renderizar sus datos
 - (O) Representación dinámica: datos de la aplicación de contenido
- (O) Funciones de plantilla: Obtener hijos - FreeMarker: Lista
 - (O) [Ejercicio] Crear el componente 'Lista de descarga de docum
- (O) Hoja de referencia: objetos de secuencia de comandos de plant

Desarrollo de sitios web

Magia de back-end de JS

(O) Material adicional LD

Herramientas de espacio

Ver comentarios insertados

Guardar para más tarde

Seguir

Compartir

(O) Funciones de plantilla: Obtener hijos - FreeMarker: Lista

Objetivos

- Para conocer la poderosa y frecuentemente utilizada **#List**directiva FreeMarker “.
- Para comprender el uso del **children(Node)** método “ de TemplatingFunctions en sus scripts de FreeMarker
- Comprender la correlación de ‘ **children(Node)**’ con la directiva FreeMarker ‘ **#list**’.

Marcador libre: **#List**

- Puede **recorrer** cualquier objeto que implemente la interfaz de la **Colección Java** .

```
[#list javaClass.getSomeCollection() as elem]
<li>${elem.property!}</li>
[/#list]
```
- Proporciona la **posibilidad** de **reaccionar** ante una **colección vacía** .
(Esto es especialmente útil para evitar etiquetas HTML vacías!

```
[#list javaClass.getSomeCollection() ]
//empty collection save
[#items as elem]
<li>${elem.title!}</li>
[/#items]
[#else]
//executed on empty collection (else is optional)
[/#list]
```

Funciones de plantillas: **children(Node)**
Recibe un nodo y devuelve una lista de sus nodos secundarios:

```
/**
 * Returns the child nodes of the provided node.
 */
public List<Node> children(Node content) throws RepositoryException {
    return content == null ? null : asNodeList(NodeUtil.getNodes(content, NodeUtil.EXCLUDE_META_DATA_FILTER));
}
```

- El **segundo parámetro** se puede utilizar para **restringir** los **elementos secundarios** devueltos por su tipo de nodo.

```
/**
 * Returns the child nodes of the provided node of a specific nodeType.
 */
public List<Node> children(Node content, String nodeName) throws RepositoryException {
    return content == null ? null : asNodeList(NodeUtil.getNodes(content, nodeName));
}
```

Correlación: **#lista** y **cmsfn.children(Node)**

- Normalmente el **children(Node)**método “ **se correlaciona** con la **#List**directiva “:

```
<ul>
[#list cmsfn.children(targetNode, "mgnl:page") as childNode]
<li><a href="${cmsfn.link(childNode)!}">${childNode.title!childNode.@name}</a></li>
[/#list]
</ul>
```
- Manejo del caso **de Colección vacía** (' **sin hijos** ' en este caso) para **evitar** etiquetas **html vacías** :

```
[#list cmsfn.children(targetNode, "mgnl:page")]
<ul>
[#items as childNode]
<li><a href="${cmsfn.link(childNode)!}">${childNode.title!childNode.@name}</a></li>
[/#items]
</ul>
[/#list]
```