

[Ejercicio] Crear componente 'teaser'

Objetivos

- Ser capaz de crear enlaces a nodos JCR.
- Ser capaz de resolver enlaces basados en UUID a nodos JCR
- Comprender cómo agregar y representar contenido desde un nodo JCR vinculado

Resultados

- Tendrá un componente **de avance** que enlaza con otras páginas y agrega contenido desde allí.

Tareas / Procedimiento

Parte 1: Crear plantilla, diálogo y guión "teaser"

1. Cree el componente **teaser** mediante CLI .
▼ [Pasos detallados](#)

⚠️ Finalmente usaremos 😊 la CLI para crear nuestro segundo componente 'teaser'.

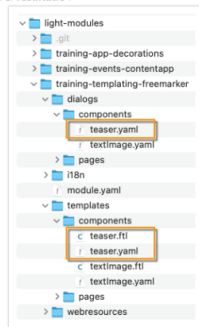
Operar la herramienta de línea de comando.

- a. Navegue en su herramienta de línea de comando hasta el **módulo de luz** `training-template-freemarker`:
- ```
/magnolia-training/magnolia-jumpstart/light-modules/training-templating-freemarker
```

- b. **Dentro del módulo de luz** *training-template-freemarker* **ejecuta** el siguiente **comando** en tu herramienta de línea de comandos:

```
mgnl create-component teaser
```

c. El resultado :



2. **Edita la definición** de la plantilla del **avance** :teaser.yaml

Operar en código VS.

- a. Navegue hasta el archivo de definición de la plantilla `:teaser.yaml`  
`/training-templating-freemarker/templates/components/teaser.yaml`

- b. **Reemplace** el código existente **con** este **código** en el `teaser.yaml` archivo:

```
title: Teaser
renderType: freemarker
dialog: training-templating-freemarker:components/teaser
templateScript: /training-templating-freemarker/templates/components/teaser.ftl
```

3. **Edita el script** de la plantilla del **avance** :teaser.ftl

Operar en código VS.

- a. **Navegue** hasta el archivo de script de la plantilla: `teaser.yaml`  
`/training-templating-freemarker/templates/components/teaser.ftl`

- b. **Reemplace** el código existente **con** este **código** en el `teaser.ftl` archivo:

```
[#assign targetPageLink = content.targetPageLink!]

[#if targetPageLink!=null]
 [#assign targetPage = cmsfn.contentById(content.targetPageLink!)]

 [#if targetPage!=null]
 <div class="default-teaser">

 [#-- Create target page link here. Use its title or page name as its text. --]

 [#-- Access abstract content of the target page. --]

 [#-- An optional possible extends: Trying also to get an image from the target page. --]

 <p>Read On</p>

 </div>
 [#elseif cmsfn.editMode]
 <div>Target Page could not be resolved.</div>
 [#if]

[#elseif cmsfn.editMode]
 <div>No target page has been chosen, please open dialog and do so.</div>
[#if]
```

4. **Edite el diálogo** del avance `:teaser.yaml`  
 ▼ [Pasos detallados](#)

Operar en código VS.

- a. **Navegue** hasta el archivo **de diálogo** de la plantilla `:teaser.yaml`  
`/training-templating-freemarker/dialogs/components/teaser.yaml`

```
label: Teaser
form:
 properties:
```

```
form:
 properties:
```

```
targetPageLink:
 $type: pageLinkField
 label: Page
 description: Choose a page to link to.
 required: true
```

5. Haga que el componente **teaser** esté disponible en el área *principal* de la **página** .
- ▼ Pasos detallados

① **Date cuenta de** dos cosas:

- Un **componente** debe estar **disponible** en **áreas** , normalmente en áreas de páginas. Como los componentes se pueden reutilizar en varias áreas de una página, como las áreas *principal* y *pie de página* , o también se pueden compartir en varias páginas diferentes. Por lo tanto, hacer que un componente esté disponible es una decisión y un proceso conscientes.
- El componente *textImage* que creamos ya se había creado para la *primera* página. Por eso no tenías que hacerlo conscientemente:

```
areas:
 main:
 type: list
 availableComponents:
 textImage:
 id: training-templating-freemarker:components/textImage
```

Operar en código VS.

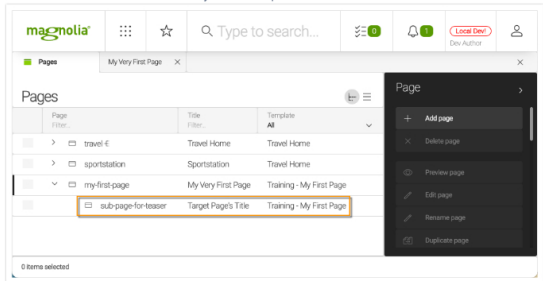
- a. **Navegue** hasta el archivo **de definición** de plantilla de la *primera* **página**: `first.yaml`  
`/training-templating-freemarker/templates/pages/first.yaml`
- b. Agregue la identificación del avance en los *Componentes disponibles* del área **principal** :

```
teaser:
 id: training-templating-freemarker:components/teaser
```

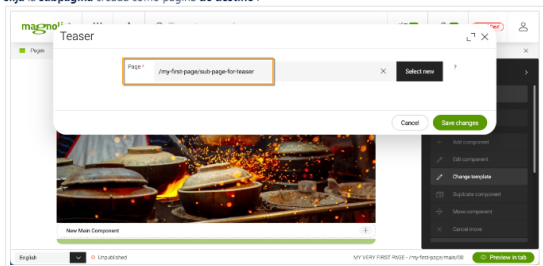
6. Pon a prueba tu trabajo.
- ▼ Pasos detallados

Operar en AdminCentral.

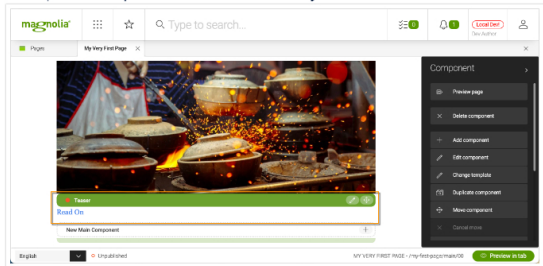
- a. **Cree una página de destino** para el avance:  
en la aplicación *Páginas* cree en su página una subpágina del mismo tipo.  
También defina un contenido de *título* y *abstracto* para ello.



- b. Abra **mi primera página** y agregue un componente **de avance** en el **área principal** :  
**elija la subpágina** creada como **página de destino** .



- c. Todo lo que se **muestra por ahora** es un enlace *Continuar leyendo* :



## Parte 2: crear un enlace a la página de destino

1. **Representa el hipervínculo** de una página de destino en el **script** de plantilla del avance `teaser.ftl`
- ▼ Pasos detallados

Operar en código VS.

- a. **Navegue** hasta el archivo **de script** de la **plantilla**: `teaser.ftl`  
`/training-templating-freemarker/templates/components/teaser.ftl`
- b. Lo primero que debe hacer es resolver el **enlace** de la **página de destino** .  
Opere debajo de este código de comentario:

```
[#-- Create target page link here. Use its title or page name as its text. --]
```

- c. Utilice el objeto **cmsfn** *TemplatingFunction* para resolver esto.  
▼ Haz trampa para el código. **PERO** primero pruébalo tú mismo!)

```
<h4 class="chapter-head">

</h4>
```

- d. Ahora **renderice** el **título de la página** de destino .  
Aplice la **verificación nula** para que cuando el título de la página **esté vacío** , tome el **nombre del nodo de la página de destino** .

① SUGERENCIA: En lugar de `$(content.nombrePropiedad !)` use `$(targetPage.nombre de la propiedad !)`.

✓ Haz trampa para el código. PERO primero pruébalo tú mismo!

```
<h4 class="chapter-head">
 $(targetPage.title!targetPage.@name)
</h4>
```

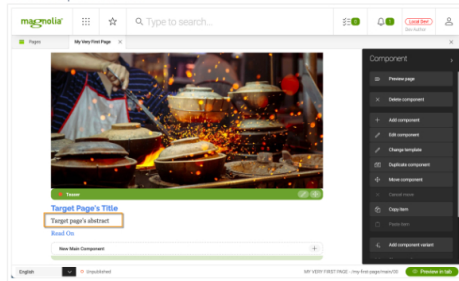
## 2. Pon a prueba tu trabajo.

✓ Pasos detallados

Operar en AdminCentral.

a. En la aplicación 'Páginas', vuelve a abrir **mi primera página**.

Observe lo que se muestra.



Es el **título** o el **nombre del nodo** de la página de destino **vinculada**.

## Parte 3: acceder al resumen de la página de destino

1. **Representa el resumen** de la página de destino en el **script** de plantilla del avance:teaser.ftl

✓ Pasos detallados

Operar en código VS.

a. **Navegue** hasta el archivo **de script de la plantilla**:teaser.yaml

/training-templating-freemarker/templates/components/teaser.ftl

b. Has accedido al título de la página en el paso anterior.

Esta vez **acceda al resumen de la página de destino**.

Opere debajo de este código de comentario:

```
[#-- Access abstract content of the target page. --]
```

① Sugerencia: en lugar de `$(content.nombrePropiedad !)` use `$(targetPage.nombre de la propiedad !)`.

✓ Haz trampa para el código. PERO primero pruébalo tú mismo!

```
<p>$(targetPage.abstract!)</p>
```

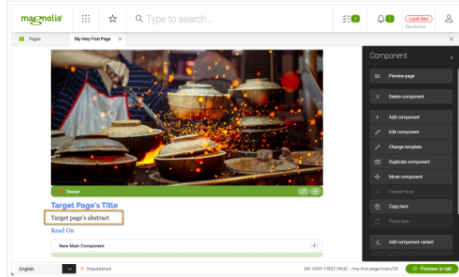
## 2. Pon a prueba tu trabajo.

✓ Pasos detallados

Operar en AdminCentral.

a. En la aplicación 'Páginas', vuelve a abrir **mi primera página**.

Observe lo que se muestra.



Es el resumen de la página **de destino**.

## Opcional: resolver el enlace de imagen de la página de destino

1. **Extensión opcional**: también renderiza la imagen de *targetPage*.

✓ Pasos detallados

a. En el guión **teaser**, el tercer comentario es una tarea opcional:

```
[#-- An optional possible extends: Trying also to get an image from the target page. --]
```

b. Intenta **resolver** esto **sólo en tu mente**. ¡No es el objetivo que lo implementes!

Si lo desea, puede intentar implementarlo **después de la capacitación**, el código de resultado está en el repositorio de git.

El objetivo principal es que pienses en posibles soluciones.

i. **Sugerencia1**: ¿Cuál es la estructura de página a la que me vinculo?

ii. **Sugerencia2**: ¿Dónde más podrían las páginas contener imágenes?

iii. **Sugerencia 3**: ¿Dónde podría tener problemas al leer propiedades de una página de destino?