

a. **Navegue** hasta el archivo **de definición** de la plantilla `:featuredCards.yaml`

/training-templating-freemarker/templates/components/cards/featuredCards.yaml

b. **Reemplace** el código existente **con este código** en el featuredCards.yamlarchivo:

```
title: Featured Cards
dialog: training-templating-freemarker:components/cards/featuredCards
templateScript: /training-templating-freemarker/templates/components/cards/featuredCards.ftl
renderType: freemarker

areas:
  cards:
    maxComponents: 4
    availableComponents:
      teaser:
        id: training-templating-freemarker:components/teaser
      textImage:
        id: training-templating-freemarker:components/textImage
```

① **Date cuenta** de tres cosas:

- Es un **componente que tiene un área**. Es la primera vez que creamos un componente que tiene un área, aquí 'tarjetas', para agregarle subcomponentes.
- Hacemos que los **dos componentes** que creamos en los ejercicios anteriores **estén disponibles** dentro del área de 'tarjetas': los componentes 'textImage' y 'teaser'.
- Hemos establecido un **máximo de componentes permitidos** en el área de 'tarjetas' de **4** usando el valor maxComponents: 4

4. **Edite el script** de plantilla de **featureCards:featuredCards.ftl**

▼ Pasos detallados

Operar en código VS.

a. **Navegue** hasta el archivo **de script de la plantilla:featuredCards.ftl**

/training-templating-freemarker/templates/components/cards/featuredCards.ftl

b. **Reemplace** el código existente **con este código** en el featuredCards.ftlarchivo:

```
[#----- ASSIGNMENTS -----]
[#assign headline = content.headline!]

[#----- RENDERING -----]

<div class="panel panel-info">
  <div class="panel-heading">
    [#if headline?has_content]
      <h3 class="panel-title">${headline}</h3>
    [#if]
  </div>

  <div class="panel-body">
    [@cms.area name="cards"/]
  </div>
</div>
```

① **Tenga en cuenta:** en el script representamos **el área de 'tarjetas'**, como estábamos acostumbrados a hacerlo en un script de página: [@cms.area name="cards"/]

5. **Edite el cuadro de diálogo** de FeatureCards:featuredCards.yaml

▼ Pasos detallados

Operar en código VS.

a. **Navegue** hasta el archivo **de diálogo** de la plantilla:featuredCards.yaml

/training-templating-freemarker/dialogs/components/cards/featuredCards.yaml

b. **Reemplace** el código existente **con este código** en el featuredCards.yamlarchivo:

```
label: Featured Card

form:
  properties:
    headline:
      $type: textfield
      label: Headline
      description: Enter a headline
```

6. Haga que el componente **featureCards** esté disponible en el área '**principal**' de la **página**.

▼ Pasos detallados

① **Tenga en cuenta:** como necesitábamos que el componente 'teaser' estuviera disponible para la página, también debemos hacer esto para el componente 'featuredCards'.

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 de las tarjetas destacadas en los '**Componentes disponibles**' del área **principal**:

```
featuredCards:
  id: training-templating-freemarker:components/cards/featuredCards
```

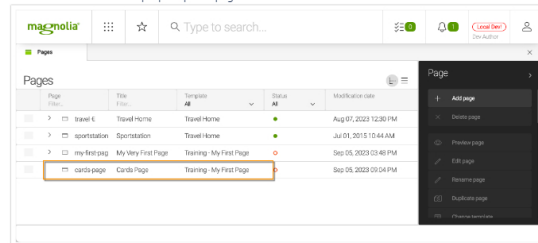
7. **Pon a prueba** tu trabajo. **crea contenido**.

▼ Pasos detallados

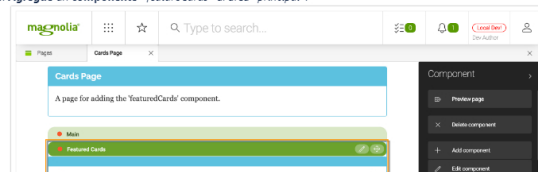
Operar en AdminCentral.

a. **Cree una nueva página** para las Tarjetas destacadas:

en la aplicación 'Páginas', cree en el nivel superior una nueva página, por ejemplo llamada '**página de tarjetas**'.
Todavía utilizamos nuestro propio tipo de página.



b. **Agregue un componente** '**featureCards**' al área '**principal**':



c. **Agregue tres componentes al área 'cartas destacadas'**: dos avances y un componente de imagen de texto (que contiene una imagen).

d. **Observe la estructura de datos** creada :
abra la aplicación del navegador ' JCR ' y observe la estructura creada de la página y los componentes dentro de:

Parte 2: Usar script de área, impactar el bucle

1. **Cree un script** de área de un nuevo componente `:featuredCardsArea.ftl`

▼ Pasos detallados

Recuerde: la plantilla de área es la única plantilla que no necesita un script de forma predeterminada.

Operar en código VS.

```
/training-templating-freemarker/templates/components/cards
```

o. Cree el guión del área: featuredCardsArea.ftl

```
/training-templating-freemarker/templates/components/cards/featuredCardsArea.ftl
```

c. Agregue el siguiente código de prueba estático :

<p>Hello World - Area Script</p>

2. **Utilice** el nuevo **script de área** dentro del área de 'tarjetas'.

▼ Pasos detallados

a. **Edite** la definición de su **componente** `featuredCards.yaml`:

```
/training-templating-freemarker/templates/components/cards/featuredCards.yml
```

b. En la configuración **del área de ' tarjetas '**, **agregue** la propiedad **templateScript** para que apunte al nuevo script del componente ' *featureCardsArea.ftl* ':

```
templateScript: /training-templating-freemarker/templates/components/cards/featuredCardsArea.ftl
```

c. El **resultado** de la configuración:

el área 'tarjetas' ahora utiliza un templateScript.

```
areas:
  cards:
    maxComponents: 4
    templateScript: ./training-templating-freemarker/templates/components/cards/featuredCardsArea.ftl
    availableComponents:
      teaser:
        id: training-templating-freemarker:components/teaser
      textImage:
        id: training-templating-freemarker:components/textImage
```

d. **Pruebe** su trabajo:

Vuelva a abrir la 'página de tarjetas':

3. Agregue **el comportamiento del script de área predeterminado** al script `featureCardsArea.ftl`.

▼ Pasos detallados

Operar en código VS.

a. Ahora agreguemos **el** comportamiento **predeterminado** de un **script de área**.

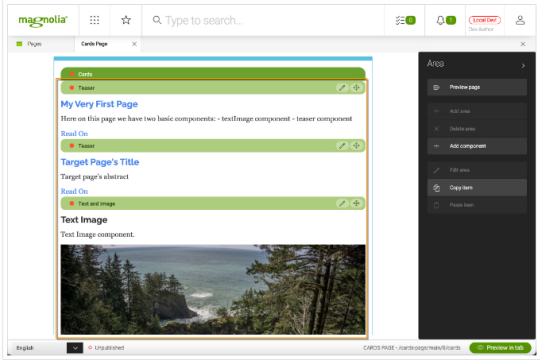
b. Navegue hasta el **script del área** :
/training-templating-freemarker/templates/components/cards/featuredCardsArea.ftl

c. Edite su **script ' featuredCards Area.ftl '** y **reemplace** el texto estático con el siguiente **código** :

```
[#list components as component ]
[@cms.component content=component /]
[/#list]
```

- ① Trate de **entender** lo siguiente:
- No hay diferencia entre no tener secuencia de comandos de área o tener una secuencia de comandos que contenga su representación predeterminada como se usa cuando no se tiene secuencia de comandos.
 - El sistema proporciona una colección denominada "componentes" que contiene todos los componentes del área misma (el sistema agrega a esta colección cada vez que agrega un nuevo nodo en "componentes disponibles").
 - La directiva de lista FreeMarker itera sobre esa colección.
 - La representación de cada componente se activa mediante la directiva " @cms.component ".

d. **Observe** lo que sucede al actualizar " *mi-primer-página* " :
no hay diferencia como cuando no tiene script.



4. **Agregue el marcado HTML** de las tarjetas de arranque de Twitter en la iteración de la lista.

↳ Pasos detallados

a. Un **caso de uso típico de los scripts de área** es controlar algún aspecto de la iteración de la lista. Agreguemos un **marcado HTML** desde Twitter Bootstrap.

b. Navegue hasta el **script del área** :
/training-templating-freemarker/templates/components/cards/featuredCardsArea.ftl

c. Edite su **script ' featuredCards Area.ftl '** y **reemplace** el texto estático con el siguiente **código** :

```
[#----- ASSIGNMENTS -----]
[#assign componentCount = components.size /]

[#switch componentCount]
[#case 2]
[#assign divColClass = "col-sm-6" /]
[#break]
[#case 3]
[#assign divColClass = "col-sm-4" /]
[#break]
[#case 4]
[#assign divColClass = "col-sm-3" /]
[#break]
[#default]
[#assign divColClass = "col-sm-12" /]
[/#switch]

[#----- RENDERING -----]

[#-- The system provides automatically to every Area script: All components of the area as a list 'components'. Therefore looping over each component
<div class="row row-cards">
  <div class="tile is-ancestor">

    [#list components as component ]
    <div class="{divColClass} tile is-parent">
      <div class="list-group-item tile is-child">
        [@cms.component content=component contextAttributes={"divColClass":divColClass} /]
      </div>
    </div>
    [/#list]

  </div>
</div>
</div>
</div>
```

① Intente **comprender** : la lógica y el marcado agregados al bucle de representación del componente. Una clave para entenderlo es la variable `{divColClass}<div class="{divColClass} tile is-parent">`

d. **Pruebe** su trabajo:
abra la aplicación " Páginas ", vuelva a abrir la " *página de tarjetas* " y observe lo que sucede:

