

magnolia

Espacios ▾ Personas Preguntas Calendarios Crear

Formación para desarrolladores de Magnolia 6.2

ÁRBOL DE PÁGINAS

Configuración y enseñanza

Desarrollo de luz

- [Configuración] Entorno de desarrollo de luz
 - Base de desarrollo ligero
 - LD: Primeros pasos
 - El mundo del autor
 - Base de arquitectura
 - LD: Fundamentos
 - Tipos de contenido
 - Creación de aplicaciones de contenido
 - Configuración de la aplicación de contenido
 - LD: Esenciales
 - [Preparación] Crear módulo de luz 'entrenamiento-aplicación-de
 - Decoradores de configuración
 - Aplicación Definiciones
 - Diseño del iniciador de aplicaciones
 - Dependencias del módulo de luz
 - [Ejercicio] Crear dependencias de módulos ligeros
 - URI virtuales
 - Anclajes YAML, incluín, heredar y anular
- Plantillas: FreeMarker
- Desarrollo de sitios web
- Magia de back-end de JS
- (O) Material adicional LD

Herramientas de espacio

Buscar

Ver comentarios insertados Guardar para más tarde Seguir Compartir

[Ejercicio] Crear dependencias de módulos ligeros

Objetivos

- Comprenda cómo definir dependencias de módulos de luz.
- Comprender el impacto en el orden de inicialización de los módulos luminosos.
- Comprenda cómo asegurarse de utilizar una determinada versión de Magnolia.

Resultados

- El módulo de luz 'training-events-contentapp' depende de Magnolia Versión 6.2 o superior .
- El módulo de luz 'training-app-decorations' depende del módulo Magnolia 'dam-app-jcr' .

Tareas / Procedimiento

1. Cree un descriptor de módulo en el módulo de luz: training-events-contentapp

Pasos detallados

Operar en código VS:

- Navegue hasta el módulo de luz 'training-events-contentapp': /training-events-contentapp
- Cree allí el archivo descriptor del módulo 'module.yaml': con este código: /training-events-contentapp/module.yaml

```
version: 1.0.0
dependencies:
  ## Ensures that this light module can only be used with Magnolia 6.2 or higher.
  core:
    version: 6.2.0/*
```

Tenga en cuenta:

este descriptor del módulo de luz garantiza que el módulo de luz 'training-events-contentapp' solo se pueda instalar/usar con Magnolia de la versión 6.2 o superior.

2. Cree un descriptor de módulo en el módulo de luz: training-app-decorations

Pasos detallados

Operar en código VS:

- Navegue hasta el módulo de luz 'decoraciones-aplicación-de-entrenamiento': /training-app-decorations
- Cree dentro del archivo descriptor del módulo 'module.yaml': con este código: /training-app-decorations/module.yaml

```
version: 1.0.0
dependencies:
  ## Ensures that this light module can only be used with Magnolia 6.2 or higher.
  core:
    version: 6.2.0/*
  ## For the exercise decorating the Assets app. Its a must: Without the decorators get loaded in a wrong order and the asset app has a strange beval
  dam-app-jcr:
    version: 3.0.0/*
```

c. Para ver los impactos de este descriptor de módulo:

- Detener Magnolia (control + c)
- Inicie Magnolia nuevamente (mgnl start): durante el inicio, vaya al siguiente paso.

3. Ver orden de carga del decorador modificado .

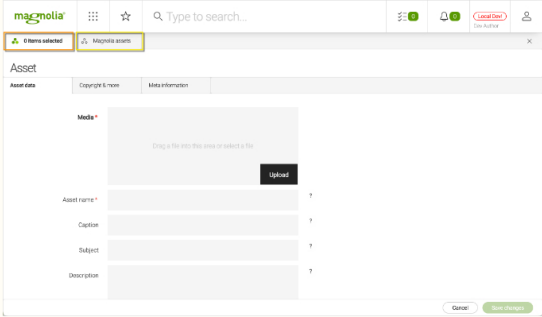
Pasos detallados

a. Recuerde el comportamiento extraño en el ejercicio:

[Ejercicio] Decorar la aplicación 'Activos': cambiar los campos del formulario de detalles

De repente se produce un comportamiento extraño al abrir la aplicación ' Activos ':

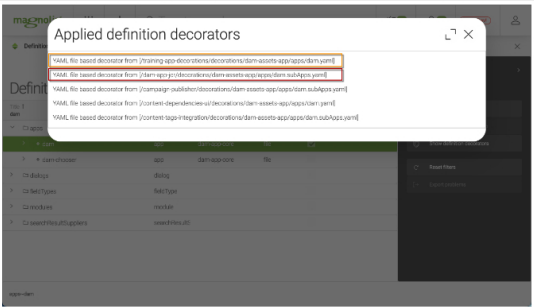
- Naranja: la aplicación Activos se abre con una subaplicación de detalles vacía etiquetada como "0 elementos seleccionados".
- Amarillo: la subaplicación del navegador es la segunda pestaña.



Resolveremos este problema en este ejercicio : [Ejercicio] Crear dependencias de módulos ligeros

b. Motivo del comportamiento extraño :

nuestro decorador (naranja) se cargó antes que el decorador principal y más importante (rojo), proporcionando la configuración básica de la subaplicación:



c. Solución para el comportamiento extraño :

el descriptor de módulo que creamos en el paso 2 definió una dependencia en el módulo 'dam-app-jcr' (rojo).

Como las dependencias de los módulos también definen su orden de inicialización, nuestro decorador (naranja) ahora se carga después.

Como los módulos se inicializan al iniciar el sistema, es necesario reiniciar el sistema (realizado en el paso 2).



d. Ver **resultado** :
Abra la aplicación ' Activos ', todo vuelve a ser normal.