

Desarrollo de páginas con Angular

Módulo 3 - Lección 2 - Lectura 01: Módulos y Servicios

En NgModules se configura el inyector del módulo, es decir, un objeto que maneja Angular internamente, y que utilizará para decidir y resolver cada vez que tenga que entregar una dependencia a un objeto.

Un NgModule es una clase marcada por el decorador @NgModule. @NgModule recibe un objeto de metadatos que describe cómo compilar la plantilla de un componente y cómo crear un inyector en tiempo de ejecución.

Los metadatos de NgModule hacen lo siguiente:

1. Declara qué componentes, directivas y tuberías (pipes) pertenecen al módulo. Hace públicos algunos de esos componentes, directivas y pipes para que las plantillas de componentes de otros módulos puedan usarlos.
2. Importa otros módulos con los componentes, directivas y pipes que necesitan los componentes en el módulo actual.
3. Proporciona servicios que los otros componentes de la aplicación pueden usar.

Cada aplicación Angular tiene al menos un módulo, el módulo raíz.

Usted arranca ese módulo para iniciar la aplicación.

Módulos de funciones o features

Los módulos de funciones (o features) son NgModules con el propósito de organizar el código.

Existen categorías en las que agrupamos a los Módulos de funciones, solo con el fin de establecer una mejor comunicación entre nosotros como programadores.

Estas categorías son las recomendadas por el equipo desarrollador de Angular, en Google, y que nos servirán para organizar mejor el código.

NgModule para Features de Dominio

Los módulos de features de dominio ofrecen una experiencia de usuario dedicada a un dominio de aplicación en particular, como editar un cliente o realizar un pedido.

Por lo general, tienen un componente superior que actúa como raíz de la característica, y los subcomponentes de soporte descienden de él.

Los módulos de features de dominio consisten principalmente en declaraciones. Solo se exporta el componente superior.

Rara vez tienen proveedores. Cuando lo hacen, la vida útil de los servicios proporcionados debe ser la misma que la vida útil del módulo.

Los módulos de features de dominio se suelen importar exactamente una vez por un módulo de funciones más grande, es decir, un módulo superior.

Pueden ser importados por el módulo de aplicación raíz de una aplicación pequeña que carece de enrutamiento.

NgModule para Enrutamiento

Un módulo de enrutamiento proporciona la configuración de enrutamiento para otro módulo y separa las preocupaciones de enrutamiento de su módulo complementario.

Un módulo de enrutamiento normalmente hace lo siguiente:

1. Define las rutas.
2. Agrega la configuración del enrutador a las importaciones del módulo.
3. Agrega proveedores de Guards y proveedores de servicio del módulo.
4. El nombre del módulo de enrutamiento debe ser paralelo al nombre de su módulo complementario, utilizando el sufijo "Enrutamiento" o "Routing". Por ejemplo, `FooModule` en `foo.module.ts` tiene un módulo de enrutamiento llamado `FooRoutingModule` en `foo-routing.module.ts`. Si el módulo complementario es el módulo de aplicación raíz, `AppRoutingModule` agrega la configuración del enrutador a sus importaciones con `RouterModule.forRoot` (rutas). Todos los demás módulos de enrutamiento son hijos que importan `RouterModule.forChild` (rutas).
5. Un módulo de enrutamiento reexporta el `RouterModule` como una convención para que los componentes del módulo complementario tengan acceso a las directivas del enrutador, como `RouterLink` y `RouterOutlet`.
6. Un módulo de enrutamiento no tiene sus propias declaraciones. Los componentes, las directivas y los pipes son responsabilidad del módulo de features, no del módulo de enrutamiento.

Un módulo de enrutamiento solo debe ser importado por su módulo complementario.

NgModule para Servicios (código de dominio)

Los módulos de servicio proporcionan servicios de utilidad tales como acceso a datos y mensajería. Idealmente, consisten enteramente de proveedores y no tienen declaraciones. El `HttpClientModule` de Angular es un buen ejemplo de un módulo de servicio.

El `AppModule` raíz es el único módulo que debe importar los módulos de servicio.

NgModule para Widgets

Un módulo de widgets hace que los componentes, directivas y pipes estén disponibles para los módulos externos. Muchas bibliotecas de componentes de interfaz de usuario de terceros son módulos de widgets.

Un módulo de widget debe consistir completamente en declaraciones, la mayoría de ellas exportadas.

Un módulo de widgets rara vez debería tener proveedores.

Importe módulos de widgets en cualquier módulo cuyas plantillas de componentes necesiten los widgets.

NgModule para enrutamiento dinámico

Los módulos de enrutamiento son módulos de funciones de dominio cuyos componentes principales son los objetivos de las rutas de navegación del enrutador.

Los módulos de funciones enrutadas no exportan nada porque sus componentes nunca aparecen en la plantilla de un componente externo.

Un módulo de características enrutadas cargado de forma perezosa no debe ser importado por ningún módulo. Hacerlo provocaría una carga voraz, anulando el propósito de la carga perezosa. Eso significa que no los verá mencionados entre las importaciones de AppModule. Otro módulo debe importar un módulo de funciones enrutadas dinámico para que el compilador conozca sus componentes.

Los módulos de funciones enrutadas rara vez tienen proveedores por los motivos explicados anteriormente. Cuando lo hacen, la vida útil de los servicios proporcionados debe ser la misma que la vida útil del módulo. No proporcione servicios singleton para toda la aplicación en un módulo de funciones enrutadas o en un módulo que importe el módulo enrutado.