

Desarrollo de páginas con Angular

Módulo 3 - Lectura 01: Introducción a la estructura de una app Angular

Vamos a analizar para empezar, desde el principio de ejecución de una aplicación, es decir lo que se llama el Bootstrapping.

Cada aplicación tiene un punto de entrada principal. Esta aplicación se creó utilizando CLI angular (que se basa en una herramienta llamada Webpack).

Cuando corremos el comando “ng serve”, ng analiza un archivo de configuración muy importante llamado angular.json para encontrar el entry point a nuestra aplicación.

Analicemos los pasos que realiza angular para encontrar los componentes y cómo comenzar la ejecución de una aplicación:

- angular.json especifica el archivo "main" es decir el principal, que en este caso actualmente es el “main.ts”
- main.ts es donde nuestra aplicación se “bootstrapea” es decir, se inicializa.
- Este proceso inicializa lo que se conoce como el módulo o ngModule, en este caso el AppModule
- Usamos el AppModule para arrancar la app. AppModule se encuentra en src/app/app.module.ts. Invito a todos que vayan analizando cada archivo ya que son claros a nivel código y seguro despierten dudas y los hagan entender muchas cosas y detalles también.
- AppModule indica el componente a usar como raíz, o de más alto nivel. En nuestro caso es AppComponent
- AppComponent tiene el tag <app-lista-destinos></app-lista-destinos> que nosotros mismos hemos realizado en la lectura anterior.

Enfoquémonos en los módulos, ya que Angular hace a los mismos componentes centrales para lograr una lógica modular en nuestras aplicaciones.

Observemos el module en cuestión:

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';
import { DestinoViajeComponent } from './destino-viaje/destino-viaje.component';
import { ListaDestinosComponent } from './lista-destinos/lista-destinos.component';

@NgModule({
```

```
declarations: [ AppComponent, DestinoViajeComponent, ListaDestinosComponent ],
imports: [
  BrowserModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }
```

Tenemos un decorador `@NgModule` que agrega metadatos a la clase inmediatamente siguiente (en este caso, `AppModule`).

Nuestro decorador `@NgModule` tiene cuatro claves: declaraciones, importaciones, proveedores y bootstrap.

Declaraciones

En la sección de declaraciones se especifica los componentes que se definen en este módulo. Puede pensar en un `NgModule` un poco como un "paquete" y las declaraciones indican qué componentes son "Propiedad de" este módulo.

Es posible que haya notado que cuando usamos `ng generate`, la herramienta agregó automáticamente nuestros componentes a esta lista de declaraciones ya que la idea es que cuando generamos un nuevo componente, la herramienta `ng` asume que queríamos que perteneciera al actual `NgModule`.

¡Esto hace que estos componentes estén disponibles para todas nuestras plantillas html dentro de este módulo!

Importaciones

Las importaciones describen qué dependencias tiene este módulo. Estamos creando una aplicación de navegador, por lo que queremos para importar el `BrowserModule`.

Más adelante veremos otros casos de imports.

Proveedores

Se utiliza proveedores para inyección de dependencia. Así que para hacer que un servicio esté disponible para ser inyectado en toda nuestra aplicación, la añadiremos aquí.

Ya daremos más información sobre inyección de dependencias.

Bootstrap

Esta sección le dice a Angular que cuando este módulo se usa para iniciar una aplicación, necesitamos cargar el componente `AppComponent` como el componente de nivel superior.