# Angular

Instalación

Tenemos que tener disponible la última versión de Node.js, que nos permitirá descargar Angular-cli. Node.js viene con npm. Angluar-cli es un herramienta en línea de comandos bastante útil para trabajar con proyectos angular2 o angular4. Angular-cli nos creará un proyecto básico o esqueleto, incluye el compilador typescript y más.

npm install –g @angular/cli@latest

Si ya lo tenemos instalado:

npm uninstall –g @angular/cli

npm cache clean

npm install –g @angular/cli@latest

Crear proyecto con angular-cli

ng new proyecto1

cd new proyecto1

ls, para ver las carpetas y ficheros generados pro angular-cli para el proyecto. 



En todas las aplicaciones Angular, existe al menos un módulo raíz, que se encuentra ubicado en el archivo app.module.ts generado por Angular CLI en el directorio src/app.

Fichero package.json

En este fichero podemos ver la versión de Node.js, del compilador /transpilador typescript.

Fichero index.html

Ubicado en /src.

Como cualquier aplicación o sitio web, los proyectos Angular disponen de un archivo index.html, ubicado en la carpeta src. Además del código habitual de un archivo html, este archivo se caracteriza por incluir la etiqueta <app-root></app-root>, que será la etiqueta del web component donde se ‘renderice’ todo el código de la aplicación. Dentro de la etiqueta raíz anterior, podemos añadir un texto que indique que la aplicación está cargando o bien un spinner, que se mostrará durante la carga de la aplicación si esta se demora. Por ejemplo: src/index.html … <app-root><p>Cargando aplicación…</p></app-root> … Si la aplicación es muy ligera, cargará tan rápido el componente raíz, que ese texto de carga, directamente no se mostrará. En la etiqueta <head></head> de este archivo, podemos añadir como en cualquier otra aplicación, CDN de librerías de fuentes e iconos así como frameworks de estilo como por ejemplo Bootstrap 4. Por ejemplo, nosotros vamos a emplear las fuentes Google y Bootstrap 4, para lo cual añadimos, dentro del <head></head>:

Fichero styles.css

Ubicado en /src.

Aquí podemos incluir todos los estilos globales, es decir para toda la aplicación. No hay que indicarlo en index.html.

Directorio assets

Ubicado en /src.

Aquí ponemos todos los fichero estáticos de la aplicación como imágnes, videos, etc..

Módulos y componentes.

Módulos

Un módulo en angular es el conjunto de código dedicado a un ámbito concreto de la aplicación o funcionalidad.

En Angular, los módulos se definen mediante una clase decorada con @NgModule. Toda aplicación de Angular tendrá al menos un módulo, el llamado módulo principal o raíz (root module), que, para el caso de aplicaciones pequeñas será único. Los módulos, se definen en archivos TypeScript y podemos decir que están compuestos de tres bloques o apartados de código.

Apartado para importación

Aquí se importan librerías y paquetes de angular así como elementos externos que se emplearán en el módulo.

Apartado para el decorador

Se emplea el decorador @NgModule, que recibe un objeto mediante el cual se configuran los metadatos:

Declarations

Las declaraciones son las llamadas vistas de un módulo. Hay 3 tipos de vistas o declaraciones, los componentes, las directivas y los pipes.

Imports

En este apartado se indican las dependencias o paquetes que empleará este módulo, cuyo origen se define en las importaciones al inicio del archivo.

Providers

Son los servicios utilizados por el módulo, disponibles para todos los componentes, y que centralizan la gestión de datos o funciones para inyectarlos en los componentes.

Bootstrap

Este metadato define la vista raíz de la aplicación y es utilizado solo por el módulo raíz. No confundir con el popular framework de estilos del mismo nombre.

Apartado para la exportación

Para dar nombre al módulo y ser exportable.

Ejemplo fichero modulo

Este ejemplo ha sido generado por angular-cli para proyecto1 (/src/app/app.module.ts):

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

import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

Componentes

Un componente controla una zona de espacio de la pantalla.

Un componente es una clase estándar de ES6 decorada con @Component.

Es equivalente en angularjs a un controlador.

En los componentes se controla la lógica de la aplicación, su vista HTML y el enlace con otros componentes.

Apartado para importación

Aquí se importan requeridos por el componente.

Apartado para el decorador

Se emplea el decorador @Component, que recibe un objeto mediante el cual se configuran los metadatos:

Selector

Define la etiqueta HTML donde se renderizará el componente.

Template

Archivo html con la vista del componente que no tiene la estructura de un archivo html, se incrustará, en este caso en el archivo index.html.

Style

Define el archivo css con los estilos del componente.

Apartado para la exportación

Para dar nombre al componente y ser exportable.

Ejemplo fichero componente

Fichero generado por angular-cli para proyecto1 /src/app/app.component.ts:

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'app';

}

Arranque del proyecto generado

ng start o ng server

Que arrancará en el puerto 4200

Para arrancar en puerto 2700: ng server –port 2700

El primer fichero typescript en ejecutarse es /src/main.ts:

import { enableProdMode } from '@angular/core';

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';

import { environment } from './environments/environment';

if (environment.production) {

enableProdMode();

}

platformBrowserDynamic().bootstrapModule(AppModule)

.catch(err => console.log(err));