

Instalación de Angular 2

Software previo

- NodeJs: Permite usar el comando npm desde la consola
- CygWin: Simulador de consola linux

Instalación Angular-cli

Para poder crear aplicaciones en Angular-cli primero hay que instalarlo desde la consola con el comando:

```
npm install -g @angular/cli
```

Existe un repositorio en GitHub con toda la info: <https://github.com/angular/angular-cli/wiki>

Crear aplicación Angular2 con angular-cli

Una vez instalado angular-cli ya se puede crear aplicaciones. Primero situarse en el directorio del cual se creará la nueva carpeta con la aplicación. Por ejemplo, si la aplicación necesita conectarse a localhost para interactuar con una base de datos con archivos PHP, lo mejor sería crear la nueva App partiendo del directorio htdocs si trabajas con xamp o www si lo haces con wamp y así evitar problemas de permisos/seguridad con llamadas a archivos php desde fuera de localhost. Si de lo contrario la App obtiene la información desde una base de datos de internet o simplemente la App no necesita conectarse a ninguna base de datos, se puede crear un nuevo proyecto angular-cli desde cualquier ubicación, por ejemplo, mis documentos.

Arrancar entorno de desarrollo

Situados previamente en la carpeta raíz de la App desde la consola cygwin, con el comando "ng serve" se arranca el servidor de test y compilador de TypeScript para ver en el navegador la aplicación en marcha. Una vez finalizado el proceso de compilado, cargando la dirección "localhost:4200" se muestra la App.

Este entorno facilita mucho las cosas, ya que cada cambio que se realice en los archivos de la App y se guarden los cambios, automáticamente el entorno de desarrollo recarga la página para mostrar los cambios.

Crear versión final de desarrollo y producción

Situados previamente en la carpeta raíz de la App.

Para crear la aplicación en modo web final (html, js y css) y poderla subir a un servidor y funcione desde cualquier navegador/usuario, se necesita crear una compilación de toda la

programación hecha con TypeScript a archivos Js, Html y Css. De eso se encarga angular-cli con el comando “ng build” para crear versión navegador en modo desarrollo y “ng build --prod” para crear versión navegador optimizada al 100%, minificando todos los archivos y creando versiones comprimidas para ser usadas por los servidores/navegadores compatibles.

Componentes

Cada página de nuestra App será programada a través de un componente.

Un componente al final va a controlar un trozo de pantalla o de la vista.

Todo lo que se puede ver en pantalla es controlado y gestionado por este tipo de elementos.

La lógica de un componente dentro una clase en Angular 2 es que da soporte a una vista interactuando con ella a través de un API con propiedades (variables) y métodos (funciones).

El componente hace de mediador entre la vista a través de la plantilla y la lógica de la app donde se incluirá el modelo de datos, un proveedor de datos externo, es decir una especie de controlador/administrador de la página en cuestión.

Crear componentes

Situados previamente en la carpeta raíz de la App desde la consola.

Angular-cli dispone de un comando para crear componentes y otros elementos de programación Angular2. En el repositorio comentado anteriormente se pueden consultar todos estos métodos integrados que facilitan el desarrollo.

Para crear componente:

ng generate component “nombre_componente”.

Para crear componente que cuelgue de otro componente

ng generate component “//nombre_componente_existente/nombre_componente”

Routing

Primero cambiar la etiqueta del template de la principal por <router-outlet></router-outlet>, ejemplo completo:

```
<h1>{{titulo}}</h1>
<nav>
  <ul>
    <li><a [routerLink]="['']">Inicio</a></li>
    <li><a [routerLink]="['contacto']">Contacto</a></li>
  </ul>
</nav>
<router-outlet></router-outlet>
<peliculas-footer></peliculas-footer>
```

Crear app.routing.ts en carpeta app

Todo el tema de rutas de la App se definen en este archivo.

```
import {ModuleWithProviders} from "@angular/core";
import {Routes, RouterModule} from "@angular/router";
```

Copiar los imports de app.module.ts de los componentes

```
const appRoutes: Routes=[
{
    path:"",
    redirectTo:'/',
    pathMatch:'full'
},
{path: "", component: PeliculasListComponent},
{path: "contacto", component: ContactoComponent}
];

export const appRoutingProviders:any[]=[];
export const routing: ModuleWithProviders= RouterModule.forRoot(appRoutes);
```

En app.module.ts

Añadir el import y provider específico del routing.

```
import {routing,appRoutingProviders} from "../app.routing"

@NgModule({
  imports:    [ BrowserModule, FormsModule, routing ],
  declarations: [ AppComponent,
                                     PeliculasListComponent,
                                     PeliculasFooterComponent,
                                     ContactoComponent,
                                     CrearPeliculaComponent
                                   ],
  providers: [appRoutingProviders],
  bootstrap: [ AppComponent ]
})
```

