

ANGULAR

Manual de uso.



Realizado por
Carmen María Soriano Tortosa

Módulo
Desalloro Web en Entorno Cliente

Índice general

1. Introducción: ¿Qué es Angular?	3
1.1. ¿Que es una aplicación SPA?	3
1.2. La historia de Angular	3
1.3. Características de Angular	3
2. Instalación de Angular	4
9. Referencias	8

Introducción: ¿Qué es Angular?

Angular es un framework de desarrollo para JavaScript creado por Google. La finalidad de este lenguaje es facilitar el desarrollo de aplicaciones web SPA y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla e óptima. Además, permite separar de forma completa el front-end y el back-end de la aplicación.

Las aplicaciones creadas en Angular pueden ser usadas en dispositivos móviles y de escritorio gracias a que es un framework cross-platform, usando el modelo vista controlador (MVC) y la ejecución es llevada por el lado del cliente, provocando así que todo dependa en gran parte del navegador.

1.1. ¿Que es una aplicación SPA?

Una web SPA(Single Page App) es una aplicación web de una sola página, en la cual la navegación entre secciones y páginas de la aplicación, así como la carga de datos, se realiza de manera dinámica, casi instantánea, asincrónicamente haciendo llamadas al servidor, y sobretodo sin resgresar la página en ningún momento.

Las aplicaciones que podemos hacer con Angular son reactivas y no recargan el navegador, todo es muy dinámico y asincrónico con ajax.

1.2. La historia de Angular

AngularJS comenzó a ser desarrollado en 2009 por Misko Hevey, siendo al principio un servicio de almacenamiento online para archivos JSON donde el precio dependía del peso de cada archivo. Tiempo después abandonó el proyecto y se lanzó angular como un proyecto open-source.

La primera versión de Angular, conocida realmente como AngularJS o Angular 1 fue lanzado en 2010, después en 2014 fue lanzada Angular2, en un principio se pensó que sería una nueva versión, pero en realidad son framework completa diferente, teniendo cada uno sus ventajas.

En 2016 se lanzó la primera versión estable de Angular2, conocida popularmente como Angular.

1.3. Características de Angular

- **Desarrollo Móvil:** El desarrollo de aplicaciones de escritorio es mucho más fácil cuando primero se manejan los problemas de rendimiento en el desarrollo móvil.
- **Modularidad:** Para desarrollar una nueva funcionalidad esta se empaqueta en un módulo, produciendo un núcleo más ligero y más rápido.
- **Compatibilidad:** Es compatible con los navegadores más modernos y recientes.

Instalación de Angular

Para poder usar Angular necesitaremos tener instalado NodeJS4, NPM3 o superior (que se instala en conjunto con NodeJS), y como recomendación, pero no obligación, GitHub y un editor de texto como puede ser Visual Estudio Code.

PASOS DE INSTALACIÓN

1. Instalamos NodeJS y rpm

2. Creamos una carpeta con el nombre que deseemos, en mi caso a sido Proyecto

3. Dentro de esa carpeta generamos los siguientes archivos:

- **package.json:** Identifica las dependencias de paquetes npm para el proyecto.

```
{
  "name": "angular-quickstart",
  "version": "1.0.0",
  "scripts": {
    "start": "tsc && concurrently \"tsc -w\" \"lite-server\" ",
    "lite": "lite-server",
    "tsc": "tsc",
    "tsc:w": "tsc -w"
  },
  "license": [
    {
      "type": "MIT",
      "url": "https://github.com/angular/angular.io/blob/master/LICENSE"
    }
  ],
  "dependencies": {
    "@angular/common": "~2.2.0",
    "@angular/compiler": "~2.2.0",
    "@angular/core": "~2.2.0",
    "@angular/forms": "~2.2.0",
    "@angular/http": "~2.2.0",
    "@angular/platform-browser": "~2.2.0",
    "@angular/platform-browser-dynamic": "~2.2.0",
    "@angular/router": "~3.2.0",
    "@angular/upgrade": "~2.2.0",
    "angular-in-memory-web-api": "~0.1.15",
    "core-js": "^2.4.1",
    "reflect-metadata": "^0.1.8",
    "rxjs": "5.0.0-beta.12",
    "systemjs": "0.19.39",
    "zone.js": "^0.6.25"
  },
  "devDependencies": {
    "@types/core-js": "^0.9.34",
    "@types/node": "^6.0.45",
    "concurrently": "^3.0.0",
    "lite-server": "^2.2.2",
    "typescript": "^2.0.3"
  }
}
```

- **tsconfig.json:** Define la configuración de TypeScript y como compilara los archivos JavaScript.

```
{
  "compilerOptions": {
    "target": "es5",
    "module": "commonjs",
    "moduleResolution": "node",
    "sourceMap": true,
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "removeComments": false,
    "noImplicitAny": false
  }
}
```

- **Systemjs.config.js:** Proporcionará la configuración de cómo serán cargados los módulos del proyecto.

```
/**
 * System configuration for Angular samples
 * Adjust as necessary for your application needs.
 */
(function (global) {
  System.config({
    paths: {
      // paths serve as alias
      'npm:': 'node_modules/'
    },
    // map tells the System loader where to look for things
    map: {
      // our app is within the app folder
      app: 'app',
      // angular bundles
      '@angular/core': 'npm:@angular/core/bundles/core.umd.js ',
      '@angular/common': 'npm:@angular/common/bundles/common.umd.js ',
      '@angular/compiler': 'npm:@angular/compiler/bundles/compiler.umd.js ',
      '@angular/platform-browser': 'npm:@angular/platform-browser/bundles/platform-brow',
      '@angular/platform-browser-dynamic': 'npm:@angular/platform-browser-dynamic/bund',
      '@angular/http': 'npm:@angular/http/bundles/http.umd.js ',
      '@angular/router': 'npm:@angular/router/bundles/router.umd.js ',
      '@angular/forms': 'npm:@angular/forms/bundles/forms.umd.js ',
      '@angular/upgrade': 'npm:@angular/upgrade/bundles/upgrade.umd.js ',
      // other libraries
      'rxjs': 'npm:rxjs ',
      'angular-in-memory-web-api': 'npm:angular-in-memory-web-api/bundles/in-memory-web',
    },
    // packages tells the System loader how to load when no filename and/or no extension
    packages: {
      app: {
        main: './main.js ',
        defaultExtension: 'js '
      },
      rxjs: {
        defaultExtension: 'js '
      }
    }
  });
})(this);
```

4. Creación de repositorio en Git Este paso es opcional.

Para crear el repositorio en Git, a través de nuestro CMD navegamos hasta la carpeta contenedora de los archivos que creamos anteriormente y procedemos a seguir los siguientes pasos:

- Comprobamos que tenemos instalado NodeJS, NPM y Git (node -v, npm -v y git --version)

```
Windows PowerShell
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion> node -v
v12.13.0
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion> npm -v
6.12.0
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion> git --version
git version 2.23.0.windows.1
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion>
```

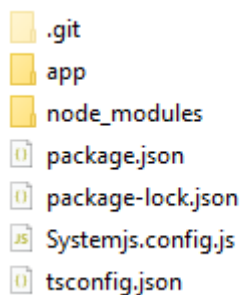
- Iniciamos un repositorio git con el comando git init.
- Agregamos nuestros archivos iniciales al repositorio con el comando git add. (Recuerda ponerle el punto al final)
- Hacemos el commit inicial con el comando git commit -m "Comentario"

```
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion> git init
Initialized empty Git repository in C:/Users/CST7/OneDrive/DAW2/Angular/Instalacion/.git/
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion> git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion> git add .
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion> git commit -m "Comentario"
[master (root-commit) 1a7a4fe] Commit inicial :)
3 files changed, 92 insertions(+)
create mode 100644 Systemjs.config.js
create mode 100644 package.json
create mode 100644 tsconfig.json
PS C:\Users\CST7\OneDrive\DAW2\Angular\Instalacion>
```

5. Instalamos los paquetes necesarios:

Para instalar los paquetes necesarios usaremos el comando npm install, tras usar este comando se nos generara una carpeta llamada node_modules dentro de nuestra carpeta. También creamos una carpeta App por lo que nuestra carpeta quedará de la siguiente manera:



6. Componentes:

Dentro de la carpeta App, creamos un archivo llamado **app.component.ts** con el siguiente contenido:

```
import { Component } from '@angular/core';
@Component({
  selector: 'my-app',
  template: '<h1>Hello Angular!</h1>'
})
export class AppComponent { }
```

7. Módulo inicial:

Dentro de un proyecto de Angular necesitaremos siempre al menos un módulo, este se encontrará dentro de la carpeta App, y lo crearemos con el nombre **app.module.ts** y con el siguiente contenido:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
@NgModule({
  imports: [ BrowserModule ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

8. Main:

Ahora crearemos nuevamente dentro de la carpeta App un archivo llamado **main.ts** con el siguiente contenido:

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from './app.module';
const platform = platformBrowserDynamic();
platform.bootstrapModule(AppModule);
```

9. Index.html

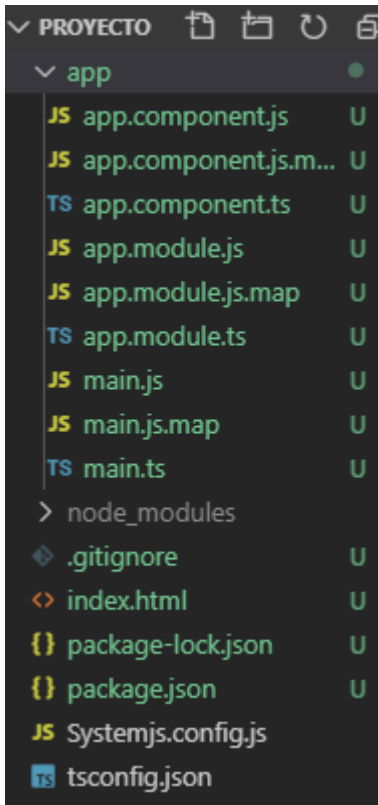
Ahora dentro de nuestra carpeta principal, en mi caso llamada Proyecto, y creamos un archivo llamado **index.html** que contenga el siguiente código: (¡Cuidado el código sigue en la siguiente página!)

```
<html>
<head>
  <title>Angular QuickStart</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- 1. Load libraries -->
  <!-- Polyfill for older browsers -->
  <script src="node_modules/core-js/client/shim.min.js"></script>
  <script src="node_modules/zone.js/dist/zone.js"></script>
  <script src="node_modules/reflect-metadata/Reflect.js"></script>
  <script src="node_modules/systemjs/dist/system.src.js"></script>
  <!-- 2. Configure SystemJS -->
  <script src="systemjs.config.js"></script>
  <script>
    System.import('app').catch(function(err){ console.error(err); });
  </script>
</head>
```

```
<!-- 3. Display the application -->
<body>
  <my-app>Loading... </my-app>
</body>
</html>
```

9. Ejecución:

Para ejecutar nuestro proyecto usaremos el comando `npm start`. Al ejecutar este comando se nos crearan los siguientes archivos:



Al ejecutarlo se nos abrirá una pestaña del navegador con la dirección localhost con el mensaje ¡Hola Angular!

Referencias

- [1] ¿CUÁNTAS HORAS ES ACONSEJABLE DORMIR SEGÚN NUESTRA EDAD? [<http://www.abc.es/familia-vida-sana/20150212/abci-dormir-horas-201502121131.html>]
- [2] JAVASCRIPT [<https://es.wikipedia.org/wiki/JavaScript>]
- [3] TYPESCRIPT [<https://es.wikipedia.org/wiki/TypeScript>]
- [4] NODE.JS [<https://es.wikipedia.org/wiki/Node.js>]
- [5] EXPRESS.JS [<https://en.wikipedia.org/wiki/Express.js>]
- [6] ANGULAR2 [<https://www.tutorialspoint.com/angular2/>]
- [7] FIREBASE [<https://en.wikipedia.org/wiki/Firebase>]**ZFZXC**
- [8] FIREBASE (GITHUB) [<https://github.com/firebase/>]
- [9] SELENIUM [<https://es.wikipedia.org/wiki/Selenium>]
- [10] JASMINE [[https://en.wikipedia.org/wiki/Jasmine_\(JavaScript_testing_framework\)](https://en.wikipedia.org/wiki/Jasmine_(JavaScript_testing_framework))]
- [11] PROTRACTOR [<https://www.adictosaltrabajo.com/tutoriales/introduccion-a-protractor/>]
- [12] VISUAL STUDIO CODE [https://en.wikipedia.org/wiki/Visual_Studio_Code]
- [13] SCRUM: THE ART OF DOING TWICE THE WORK IN HALF THE TIME [Autor: Jeff Sutherland]
- [14] UML DISTILLED [Autor: Martin Fowler]
- [15] TRAVIS [https://en.wikipedia.org/wiki/Travis_CI]
- [16] SPRINT [<https://proyectosagiles.org/ejecucion-iteracion-sprint/>]
- [17] ¿CADA CUÁNTO SOLEMOS RENOVAR EL MÓVIL EN ESPAÑA? [<http://byemovil.com/renovar-el-movil>]
- [18] UNDERSTANDING MODEL-VIEW-CONTROLLER [Autor: Stefano Borini <https://www.gitbook.com/book/stefanoborini/modelviewcontroller/details>]