



ANGULAR

Qué es Angular y
sus ventajas. Cómo
se instala.

Coding Potions –lección 1

Qué es Angular

Angular es un framework para la creación de páginas web SPA mantenido por Google. SPA es el acrónimo de 'Single Page Application' o lo que es lo mismo, cuando un usuario entra en una web SPA, se carga todo a la vez en una misma página y Angular lo que hace por debajo es cambiar la vista al navegar por la página para que de la apariencia de una web normal. ¿Qué ventajas tiene?

- Velocidad de carga lenta la primera vez que se abre la web, pero luego navegar por la web es instantáneo debido a que se carga toda la web de golpe.
- Cómo SPA es una página solo hay una ruta que tiene que enviar el servidor.
- Aplicaciones modulares y escalares.

Ventajas de Angular

- Lenguaje Typescript, tiene una sintaxis muy parecida a Java, con tipado estático.
- Sigue el patrón MVC, con la vista separada de los controladores.
- Basado en componentes, es decir, podemos escribir componentes web con vista y lógica para después reutilizarlos en otras páginas.
- Comunidad muy grande con multitud de tutoriales y librerías.

- Inyección de dependencias, un patrón de diseño que se basa en pasar las dependencias directamente a los objetos en lugar de crearlas localmente.
- Programación reactiva, la vista se actualiza automáticamente a los cambios.
- Dispone de asistente por línea de comandos para crear proyectos base (Angular cli).

- Se integra bien con herramientas de testing.
- Se integra bien con Ionic, para adaptar aplicaciones web a dispositivos móviles.

Qué es mejor Angular o Jquery

Si ya conoces Jquery seguramente quieres ver para qué sirve Angular, por si te interesa cambiarte. La realidad es que

aunque JQuery se sigue utilizando mucho, está en desuso. En los tiempos en los que vivimos, los desarrolladores frontend se están pasando a otros frameworks como Angular o React. El por qué de esto radica en que estos frameworks ofrecen un conjunto de herramientas completas para hacer una web. Es decir, mientras que JQuery te da ciertas facilidades y ahorras código Javascript, con Angular, por ejemplo, tienes a tu disposición muchas más cosas, como por ejemplo, formas de

definir rutas para la web, reactividad en las vistas, protección de rutas, etc.

Como conclusión diría que JQuery viene bien para proyectos muy sencillos si quieres trabajar con Javascript sin más, pero a la hora de hacer una web completa te recomiendo un framework completo como Angular o ReactJS

Cómo instalar Angular

Para instalarlo tenemos que disponer de Node y NPM instalados en el equipo. Si no

lo tienes instalado, puedes decargar las dos herramientas desde aquí:

<https://www.npmjs.com/get-npm>.

Una vez instalado NPM ejecuta la terminal y escribe:

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

Este comando instalará Angular cli de forma global en nuestro equipo. Angular cli es la herramienta de consola de Angular que nos ayudará en la programación con Angular.

Para que Angular cli cree un proyecto vacío de base para crear una aplicación con Angular, escribe:

```
ng new NOMBRE_APP
```

Cambia NOMBRE_APP por el nombre que le quieras dar a tu aplicación.

Ahora muévete en la terminal con el comando `cd` a la carpeta que se acaba de crear y ejecuta:

```
npm install
```

Este comando servirá para que se instalen en el proyecto las dependencias que hagan falta.

Por último para ejecutar la aplicación web que acabamos de crear simplemente:

```
ng serve --open
```

El flag `--open` sirve para que se abra automáticamente el navegador web con la web. Por defecto Angular se ejecuta en el puerto 4200.

Si todo ha ido bien, veremos una web como ésta (si no se te ha abierto el navegador, abre <http://localhost:4200/>):



Angular cli también permite compilar una web para subirla a producción, para hacerlo introduce este comando:

`ng build`

De esta forma Angular se encargará de comprimir todos los archivos Typescript en archivos Javascript entendibles por el navegador.

Cómo es la estructura de una app creada con Angular

Cuando generamos un proyecto con Angular cli nos genera la siguiente estructura (en mi caso he llamado a la aplicación tutoApp:

└ TUTOAPP

└ e2e

└ node_modules

└ src

└ app

app.component.css

app.component.html

app.component.spec.ts

app.component.ts

app.module.ts

└ assets

└ environments

favicon.ico

index.html

main.ts

polyfills.ts

styles.css

test.ts

tsconfig.app.json

tsconfig.spec.json

typings.d.ts

Voy a pasar a explicar por encima para que sirve cada archivo y carpeta:

- e2e: Esta carpeta por el momento no nos es útil, aquí se encuentra el código para escribir tests end to end que prueben la aplicación
- node_modules: En esta carpeta se encuentran las librerías de angular y sus dependencias, cuando instalemos librerías se añadirán aquí. Generalmente no hay que tocar nada de esta carpeta.

- src: Aquí se encuentran los archivos que componen nuestra aplicación
 - app: Aquí se donde se van a encontrar los componentes, vistas, y servicios de la app.

Por el momento hay un componente llamado app con sus respectivos archivos (css, html controlador, tests, etc)
 - app.module.ts: En este archivo se especifica los componentes que

vamos a usar en la app web. Cuando creamos un componente tenemos que importarlo en este archivo.

- favicon: El favicon de la web
- index.html: Punto de entrada a nuestra web, este archivo se carga en todas las webs, por lo que puedes poner código para que se incluya en todas las vistas.

- `main.ts`: Algunas configuraciones de Angular, de momento no nos hace falta tocarlo.
- `polyfills.ts`: Configuraciones y código que se ejecutará antes de que se inicie la app. De momento tampoco nos hace falta tocarlo.
- `styles.css`: Estilos css globales que se aplicarán en toda las vistas de la página.

- test.ts: Configuración para los tests. No es útil de momento.
- tsconfig.app.json, tsconfig.spec.json y typings.d.ts: Lo mismo que el anterior.
- .angular-cli.json: Archivo de configuración de la app.
- .editorconfig: Configuraciones a la hora de desarrollar, por ejemplo, como van a ser las identaciones.

- `.gitignore`: Archivo para que git ignore ciertas carpetas que no hace falta subir, como `node_modules` (cuando te bajas el proyecto ejecutas `npm install` para que descargue las dependencias en `node_modules`).
- `karma.conf.js`: Más configuraciones para los tests, esta vez los de Karma.
- `package-lock.json`: Árbol de dependencias que se crea automáticamente

- `package.json`: Archivo con las dependencias instaladas y los comandos que se pueden ejecutar con npm
- `protractor.conf.js`: Configuración para protractor, una herramienta para realizar tests en el navegador.
- `README.md` Archivo readme con información de la aplicación.
- `tsconfig.json`: Configuración para Typescript, el lenguaje de Angular.

- tslint.json: Configuración del linter de TypeScript (un linter sirve para hacer comprobaciones del estilo del código que escribimos).

Ahora si abres el archivo `app.component.ts` situado en la carpeta `src/app` y cambias el string de:

```
title = 'app';
```

por:

```
title = 'my wonderful app';
```

por poner un ejemplo, si ahora abres la página (si no tienes funcionando el comando `ng serve`, ejecútalo), verás que ahora en la página pone `Welcome to my wonderful app!`

Como ves, existe una variable llamada `title` (Typescript tiene inferencia de tipos y no hace falta que especifiques de que tipo es la variable) que automáticamente se pinta en el `html`, para ello si abres el archivo `app.component.html` verás que en la 4ª línea hay:

Welcome to {{ title }}!

Con {{ nombre_variable }} puedes pintar variables creadas en el controlador (archivos ts) de su correspondiente componente (en este caso el componente es app).

El mejor IDE para programar en Angular

Sin duda te recomiendo que instales el [vscode](#). Si no lo conocías, es un editor de textos muy avanzado open source mantenido por Microsoft. Es muy recomendable porque en los últimos tiempos ha mejorado mucho, es ligero y muy adaptable a los requisitos de todo el mundo mediante extensiones. Además cuenta con una comunidad en auge, por lo

que es una buena opción a futuro. Te recomiendo si lo instalas que eches un vistazo a [esta extensión](#) pensada para Angular, ofrece más soporte, snippets, iconos, etc.

Conclusiones

Recapitulando, hemos visto qué es Angular, como instalarlo, cómo crear el esqueleto de una app, y una idea aproximada de para qué sirven los archivos y carpetas que crea por defecto. Como he dicho si que te recomiendo a que aprendas Angular porque es una apuesta segura de futuro. Con Angular vas a aprender además, Typescript, un lenguaje de programación

que aporta muchas ventajas respecto a Javascript plano.

Te animo a que pruebes y cambies cosas del código para que vayas viendo como funciona Angular y Typescript. Si quieres encontrar más info de Angular lo puedes hacer en su página oficial:

<https://angular.io/>.
