

Instalación Angular con Angular CLI

Angular CLI es una herramienta que nos hace más fácil el proceso de instalación de Angular y agiliza el inicio de un proyecto con este framework de Google.

Importante: Angular CLI no es Angular, es una herramienta para trabajar con Angular. El framework Angular no se instala en nuestro equipo sino que cada vez que creemos un nuevo proyecto se instalarán los paquetes de Angular, a través de *npm*, en nuestro propio proyecto.

¿Qué nos ofrece Angular CLI?

1. Instalación y esqueleto básico de Angular
2. Comandos para la generación de código(componentes, servicios, etc)
3. Herramientas de minificación y builds para las webapps
4. Herramientas de Testing y Debugging

Para instalar Angular CLI lanzamos el siguiente comando (la opción `-g` para que se instale de manera global y podamos acceder desde cualquier ruta del equipo)

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

Nota: Si ya tenemos una instalación anterior de Angular CLI seguir los pasos de esta URL:

<https://github.com/angular/angular-cli#updating-angular-cli>

Ahora vamos a **crear una aplicación** esqueleto base de Angular (instalar Angular):

```
ng new <nombre_proyecto>
```

En este momento tenemos un proyecto instalado y a partir de ahora ya podemos **escribir nuestro código dentro del directorio 'src'**.

Para **lanzar la aplicación en un servidor web**, como el que típicamente usábamos en cualquier app de Angular 2 estándar. Usaremos el comando:

```
ng serve
```

Importante: El comando anterior hay que ejecutarlo dentro de la carpeta de un proyecto angular-cli, es decir, dentro del proyecto que hemos creado al principio (<nombre_proyecto>).

Ahora tendremos disponible la webapp de Angular 2 en <http://localhost:4200/>

Nota: Para lanzar el proyecto/app en otro puerto diferente al 4200 ejecutar el comando

```
ng serve --port 4201 (p.e.)
```

Podemos generar algunos **componentes del framework**; veamos algunos comandos para hacer esto.

- Generar un **componente**:

```
ng generate component  
<nuevo-componente>
```

o también

```
ng g c <nuevo-
```

componente>

- Generar un componente omitiendo el archivo para pruebas 'app.component.spec.ts':

```
ng g c <nuevo_comp> --spec false
```

- Generar un **pipe**:

```
ng generate pipe nuevo-filtro
```

- Generar un **servicio**:

```
ng generate service nuevo-servicio
```

The Application Shell

Install the Angular CLI

Install the [Angular CLI](#), if you haven't already done so.

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

Create a new application

Create a new project named `angular-tour-of-heroes` with this CLI command.

```
ng new angular-project
```

The Angular CLI generated a new project with a default application and supporting files.

Serve the application

Go to the project directory and launch the application.

```
cd angular-project
```

```
ng serve --open
```

The `ng serve` command builds the app, starts the development server, watches the source files, and rebuilds the app as you make changes to those files.

The `--open` flag opens a browser to `http://localhost:4200/`.

You should see the app running in your browser.

Angular components

The page you see is the *application shell*. The shell is controlled by an Angular component named `AppComponent`.

Components are the fundamental building blocks of Angular applications. They display data on the screen, listen for user input, and take action based on that input.

Change the application title

Open the project in your favorite editor or IDE and navigate to the `src/app` folder.

You'll find the implementation of the shell `AppComponent` distributed over three files:

1. `app.component.ts`— the component class code, written in TypeScript.
2. `app.component.html`— the component template, written in HTML.
3. `app.component.css`— the component's private CSS styles.

Open the component class file (`app.component.ts`) and change the value of the `title` property to 'Tour of Heroes'.

`app.component.ts` (class title property)

```
title = 'Tour of Heroes';
```

Open the component template file (`app.component.html`) and delete the default template generated by the Angular CLI. Replace it with the following line of HTML.

`app.component.html` (template)

```
<h1>{{title}}</h1>
```

The double curly braces are Angular's *interpolation binding* syntax. This interpolation binding presents the component's `title` property value inside the HTML header tag.

The browser refreshes and displays the new application title.

Add application styles

Most apps strive for a consistent look across the application. The CLI generated an empty `styles.css` for this purpose. Put your application-wide styles there.

Here's an excerpt from the `styles.css` for the *Tour of Heroes* sample app.

`src/styles.css` (excerpt)

```
/* Application-wide Styles */

h1 {

  color: #369;

  font-family: Arial, Helvetica, sans-serif;

  font-size: 250%;

}

h2, h3 {

  color: #444;

  font-family: Arial, Helvetica, sans-serif;
```

```

    font-weight: lighter;

}

body {

    margin: 2em;

}

body, input[text], button {

    color: #888;

    font-family: Cambria, Georgia;

}

/* everywhere else */

* {

    font-family: Arial, Helvetica, sans-serif;

}

```

Final code review

The source code for this tutorial and the complete *Tour of Heroes* global styles are available in the [live example](#) / [download example](#).

Here are the code files discussed on this page:

```

src/app/app.component.ts
src/app/app.component.html
src/styles.css (excerpt)

```

```

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

@Component({

    selector: 'app-root',

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

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

})

export class AppComponent {

    title = 'Tour of Heroes';

```

}

Summary

- You created the initial application structure using the Angular CLI.
- You learned that Angular components display data.
- You used the double curly braces of interpolation to display the app title.