

Create a new project



Contents >

Set up your environment

Create a new workspace and an initial application

Serve the application

...

You begin by creating an initial application using the Angular CLI. Throughout this tutorial, you'll modify and extend that starter application to create the Tour of Heroes app.

In this part of the tutorial, you'll do the following:

1. Set up your environment.
2. Create a new workspace and initial app project.
3. Serve the application.
4. Make changes to the application.

For the sample app that this page describes, see the [live example](#) / [download example](#).

Set up your environment

To set up your development environment, follow the instructions in [Local Environment Setup](#).

Create a new workspace and an initial application

You develop apps in the context of an Angular [workspace](#). A workspace contains the files for one or more [projects](#). A project is the set of files that comprise an app, a library, or end-to-end (e2e) tests. For this tutorial, you will create a new workspace.

To create a new workspace and an initial app project:

1. Ensure that you are not already in an Angular workspace folder. For example, if you have previously created the Getting Started workspace, change to the parent of that folder.
2. Run the CLI command `ng new` and provide the name `angular-tour-of-heroes`, as shown here:

```
ng new angular-tour-of-heroes
```



3. The `ng new` command prompts you for information about features to include in the initial app project. Accept the defaults by pressing the Enter or Return key.

The Angular CLI installs the necessary Angular npm packages and other dependencies. This can take a few minutes.

It also creates the following workspace and starter project files:

- A new workspace, with a root folder named `angular-tour-of-heroes`.
- An initial skeleton app project, also called `angular-tour-of-heroes` (in the `src` subfolder).
- An end-to-end test project (in the `e2e` subfolder).
- Related configuration files.

The initial app project contains a simple Welcome app, ready to run.

Serve the application

Go to the workspace directory and launch the application.

```
cd angular-tour-of-heroes
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.

Make changes to the application

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

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.

Change the application title

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.

Open `src/styles.css` and add the code below to the file.

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[type="text"], button {
  color: #333;
  font-family: Cambria, Georgia;
}
```



```
/* everywhere else */
* {
  font-family: Arial, Helvetica, sans-serif;
}
```

Final code review

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.

RESOURCES

About
Resource Listing
Press Kit
Blog
Usage Analytics

HELP

Stack Overflow
Gitter
Report Issues
Code of Conduct

COMMUNITY

Events
Meetups
Twitter
GitHub
Contribute

LANGUAGES

简体中文版
正體中文版
日本語版
한국어

Super-powered by Google ©2010-2020. Code licensed under an MIT-style License. Documentation licensed under CC BY 4.0.

Version 10.0.10-local+sha.b32126c335.

