**Angular Cli 2 App**

[**https://update.angular.io/**](https://update.angular.io/)**: for understanding ng versions and upgrate or degrade.**

One of the easiest ways to start a new Angular 2 application is to use Angular’s [command-line interface](https://github.com/angular/angular-cli) (CLI).

1. Create directory on your system by using cmd

🡪$ mkdir ang-cli

1. Then redirect to ang-cli folder
2. Then install Angular CLI, run:

* $ npm install -g angular-cli

1. Which will install the ng command globally on your system.

To verify whether your installation completed successfully, you can run:

* $ ng version

1. Then create new app by using following command

* $ ng new app-name (ng new test-app)

1. Then redirect to app folder (test-app)
2. Run project by

🡪$ng serve

1. To create a new component refer the following command

ng generate component hero-detail

1. For routing in angular 2

ng generate module app-routing --flat --module=app

**Folder Structure**

* The src folder

Your app lives in the src folder. All Angular components, templates, styles, images, and anything else your app needs go here. Any files outside of this folder are meant to support building your app.

Src

app

app.component.css

app.component.html

app.component.spec.ts

app.component.ts

app.module.ts

assets

. gitkeep

environments

environment.prod.ts

environment.ts

favicon.ico

index.html

main.ts

polyfills.ts

styles.css

test.ts

tsconfig.app.json

tsconfig.spec.json

|  |  |
| --- | --- |
| **File** | **Purpose** |
| app/app.component.{ts,html,css,spec.ts} | Defines the AppComponent along with an HTML template, CSS stylesheet, and a unit test. It is the **root**component of what will become a tree of nested components as the application evolves. |
| app/app.module.ts | Defines AppModule, the [root module](https://angular.io/guide/bootstrapping) that tells Angular how to assemble the application. Right now it declares only the AppComponent. Soon there will be more components to declare. |
| assets/\* | A folder where you can put images and anything else to be copied wholesale when you build your application. |
| environments/\* | This folder contains one file for each of your destination environments, each exporting simple configuration variables to use in your application. The files are replaced on-the-fly when you build your app. You might use a different API endpoint for development than you do for production or maybe different analytics tokens. You might even use some mock services. Either way, the CLI has you covered. |
| favicon.ico | Every site wants to look good on the bookmark bar. Get started with your very own Angular icon. |
| index.html | The main HTML page that is served when someone visits your site. Most of the time you'll never need to edit it. The CLI automatically adds all js and css files when building your app so you never need to add any <script> or <link> tags here manually. |
| main.ts | The main entry point for your app. Compiles the application with the [JIT compiler](https://angular.io/guide/glossary#jit) and bootstraps the application's root module (AppModule) to run in the browser. You can also use the [AOT compiler](https://angular.io/guide/aot-compiler) without changing any code by appending the--aot flag to the ng build and ng serve commands. |
| polyfills.ts | Different browsers have different levels of support of the web standards. Polyfills help normalize those differences. You should be pretty safe with core-js and zone.js, but be sure to check out the [Browser Support guide](https://angular.io/guide/browser-support) for more information. |
| styles.css | Your global styles go here. Most of the time you'll want to have local styles in your components for easier maintenance, but styles that affect all of your app need to be in a central place. |
| test.ts | This is the main entry point for your unit tests. It has some custom configuration that might be unfamiliar, but it's not something you'll need to edit. |
| tsconfig.{app|spec}.json | TypeScript compiler configuration for the Angular app (tsconfig.app.json) and for the unit tests (tsconfig.spec.json). |

### The root folder

The src/ folder is just one of the items inside the project's root folder. Other files help you build, test, maintain, document, and deploy the app. These files go in the root folder next to src/.

my-app

e2e

app.e2e-spec.ts

app.po.ts

tsconfig.e2e.json

node\_modules/...

src/...

.angular-cli.json

.editorconfig

.gitignore

karma.conf.js

package.json

|  |  |
| --- | --- |
| **File** | **Purpose** |
| e2e/ | Inside e2e/ live the end-to-end tests. They shouldn't be inside src/ because e2e tests are really a separate app that just so happens to test your main app. That's also why they have their own tsconfig.e2e.json. |
| node\_modules/ | Node.js creates this folder and puts all third party modules listed in package.json inside of it. |
| .angular-cli.json | Configuration for Angular CLI. In this file you can set several defaults and also configure what files are included when your project is built. Check out the official documentation if you want to know more. |
| .editorconfig | Simple configuration for your editor to make sure everyone that uses your project has the same basic configuration. Most editors support an .editorconfig file. See [http://editorconfig.org](http://editorconfig.org/) for more information. |
| .gitignore | Git configuration to make sure autogenerated files are not commited to source control. |
| karma.conf.js | Unit test configuration for the [Karma test runner](https://karma-runner.github.io/), used when running ng test. |
| package.json | npm configuration listing the third party packages your project uses. You can also add your own [custom scripts](https://docs.npmjs.com/misc/scripts)here. |