Setup for local development

{@a develop-locally}

The QuickStart live-coding example is an Angular *playground*. It's not where you'd develop a real application. You should develop locally on your own machine ... and that's also how we think you should learn Angular.

Setting up a new project on your machine is quick and easy with the **QuickStart seed**, maintained on github.

Make sure you have node and npm installed.

{@a clone}

Clone

Perform the *clone-to-launch* steps with these terminal commands.

git clone https://github.com/angular/quickstart.git quickstart cd quickstart npm install npm start

`npm start` fails in _Bash for Windows_ in versions earlier than the Creator's Update (April 2017).

{@a download}

Download

Download the QuickStart seed and unzip it into your project folder. Then perform the remaining steps with these terminal commands.

cd quickstart npm install npm start

`npm start` fails in _Bash for Windows_ in versions earlier than the Creator's Update (April 2017).

{@a non-essential}

Delete *non-essential* files (optional)

You can quickly delete the *non-essential* files that concern testing and QuickStart repository maintenance (***including all git-related artifacts*** such as the `.git` folder and `.gitignore` !).

Do this only in the beginning to avoid accidentally deleting your own tests and git setup!

Open a terminal window in the project folder and enter the following commands for your environment:

OS/X (bash)

xargs rm -rf < non-essential-files.osx.txt rm src/app/*.spec*.ts rm non-essential-files.osx.txt

Windows

for /f %i in (non-essential-files.txt) do del %i /F /S /Q rd .git /s /q rd e2e /s /q

{@a seed}

What's in the QuickStart seed?

The **QuickStart seed** contains the same application as the QuickStart playground. But its true purpose is to provide a solid foundation for *local* development. Consequently, there are *many more files* in the project folder on your machine, most of which you can learn about later.

{@a app-files}

Focus on the following three TypeScript (`.ts`) files in the **`/src`** folder.

src
app
app.component.ts
app.module.ts
main.ts

All guides and cookbooks have *at least these core files*. Each file has a distinct purpose and evolves independently as the application grows.

Files outside `src/` concern building, deploying, and testing your app. They include configuration files and external dependencies.

Files inside `src/` "belong" to your app. Add new Typescript, HTML and CSS files inside the `src/` directory, most of them inside `src/app`, unless told to do otherwise.

The following are all in `src/`

File	Purpose
`app/app.component.ts`	Defines the same `AppComponent` as the one in the QuickStart playground. 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](guide/bootstrapping "AppModule: the root module") that tells Angular how to assemble the application. Right now it declares only the `AppComponent`. Soon there will be more components to declare.
`main.ts`	Compiles the application with the [JIT compiler](guide/glossary#jit) and [bootstraps](guide/bootstrapping#main "bootstrap the application") the application's main module (`AppModule`) to run in the browser. The JIT compiler is a reasonable choice during the development of most projects and it's the only viable choice for a sample running in a _live-coding_ environment like Plunker. You'll learn about alternative compiling and [deployment](guide/deployment) options later in the documentation.

Next Step If you're new to Angular, we recommend you follow the [tutorial](tutorial "Tour of Heroes tutorial").

{@a install-prerequisites}

Appendix: node and npm

Node.js and npm are essential to modern web development with Angular and other platforms. Node powers client development and build tools. The *npm* package manager, itself a *node* application, installs JavaScript libraries.

Get them now if they're not already installed on your machine.

Verify that you are running node `v4.x.x` or higher and npm `3.x.x` or higher by running the commands `node -v` and `npm -v` in a terminal/console window. Older versions produce errors.

We recommend nvm for managing multiple versions of node and npm. You may need nvm if you already have

projects running on your machine that use other versions of node and npm.

{@a why-locally}

Appendix: Why develop locally

Live coding in the browser is a great way to explore Angular.

Links on almost every documentation page open completed samples in the browser. You can play with the sample code, share your changes with friends, and download and run the code on your own machine.

The QuickStart shows just the `AppComponent` file. It creates the equivalent of `app.module.ts` and `main.ts` internally *for the playground only*. so the reader can discover Angular without distraction. The other samples are based on the QuickStart seed.

As much fun as this is ...

- you can't ship your app in plunker
- you aren't always online when writing code
- transpiling TypeScript in the browser is slow
- the type support, refactoring, and code completion only work in your local IDE

Use the live coding environment as a *playground*, a place to try the documentation samples and experiment on your own. It's the perfect place to reproduce a bug when you want to file a documentation issue or file an issue with Angular itself.

For real development, we strongly recommend developing locally.