



## BTI425

Web Programming for Apps and Services

[Notes](#)[Weekly](#)[Resources](#)[Graded work](#)[Policies](#)[Standards](#)[Professors & addendum](#)[Code examples](#)

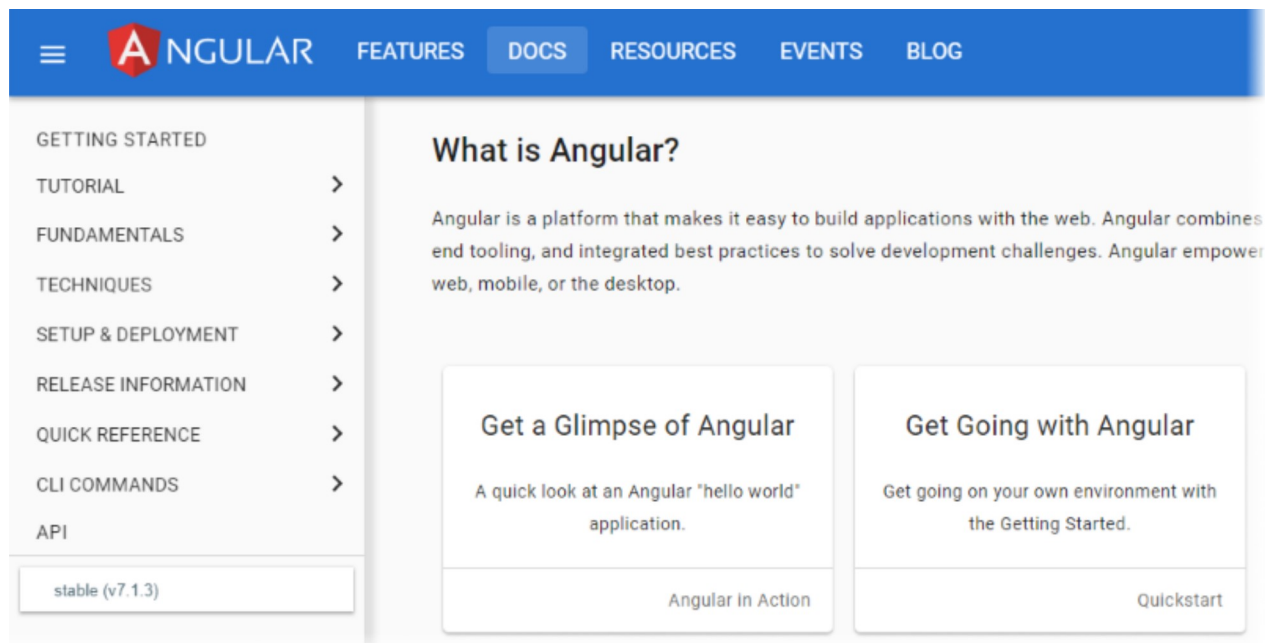
---

### Getting started with Angular

The next part of our learning journey will depend on the excellent content that the Angular development team has published.

Get ready by visiting the [Angular home page](#), and read/scan the following content:

- Landing page
- FEATURES page
- The DOCS landing page
- A few screenfuls of the GETTING STARTED page



## Installation (setup)

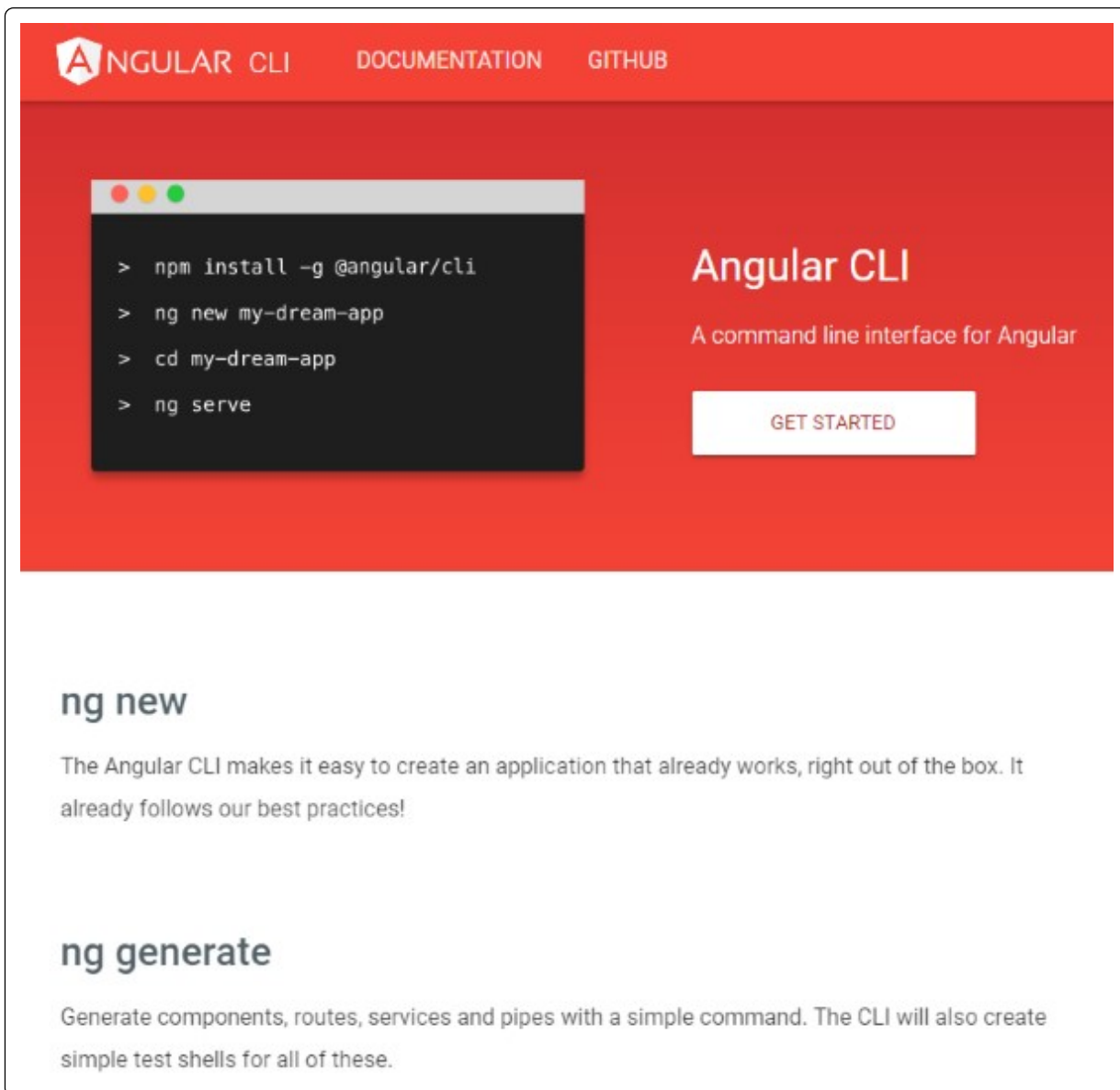
As described in the SETUP notes, the [Angular CLI](#) package is used to create projects, generate code, and enable development. It adds the `ng` command to your shell.

Install Angular CLI on your computer with this command:

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

The [Angular CLI documentation](#) explains how to do many tasks, including:

- [Create a new](#) Angular project (app)
- Generate the code files for a [new component](#)
- [Run the app](#) on localhost



## Create an app

Next, create a new app. Assuming that you want to create a new app (and folder) named “my-dream-app”:

```
# upper and lower case use for the options is important
ng new my-app-name --routing -S -g
```

The process will create a new folder, with the code needed to get started.

### Notes:

*The --routing option adds the code we need for “routing”, which is a topic that will be covered soon. Adding routing now (when the new project is created) is a best practice. The -S option avoids adding “testing” code. We don’t need that in the near future.*

*The `-g` option does not create a Git repository for the project. We don't need a repo in the near future.*

## Run the app

Similar to a React app, an Angular app is a client-side front-end app. It does NOT have a server part to it. A browser user “pulls” an Angular app, by visiting a URL that is the “root” of the app. As you will learn later, the build, packaging, and deployment process creates a bundle of files that are sent to the browser when the root of the app is requested for the first time.

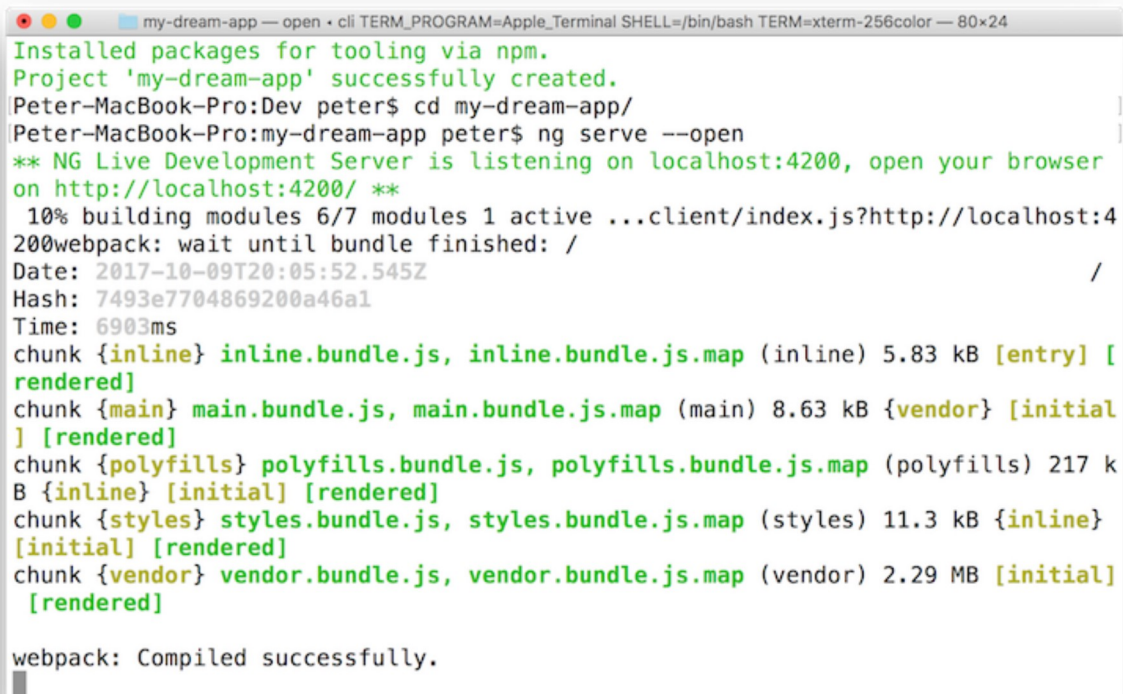
That being said, during development on your computer, will start an on-demand per-instance Node.js server, and listen on the app's URL. Then, from a browser, the app is pulled to the browser's memory, and is ready for use.

Therefore, start the server listener:

```
cd my-app-name
ng serve --open
```

*Note: The `--open` option opens a browser tab and loads the app.*

The server begins listening on HTTP port 4200.



```
my-dream-app — open • cli TERM_PROGRAM=Apple_Terminal SHELL=/bin/bash TERM=xterm-256color — 80x24
Installed packages for tooling via npm.
Project 'my-dream-app' successfully created.
Peter-MacBook-Pro:Dev peter$ cd my-dream-app/
Peter-MacBook-Pro:my-dream-app peter$ ng serve --open
** NG Live Development Server is listening on localhost:4200, open your browser
on http://localhost:4200/ **
10% building modules 6/7 modules 1 active ...client/index.js?http://localhost:4
200webpack: wait until bundle finished: /
Date: 2017-10-09T20:05:52.545Z /
Hash: 7493e7704869200a46a1
Time: 6903ms
chunk {inline} inline.bundle.js, inline.bundle.js.map (inline) 5.83 kB [entry] [
rendered]
chunk {main} main.bundle.js, main.bundle.js.map (main) 8.63 kB {vendor} [initial
] [rendered]
chunk {polyfills} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 217 k
B {inline} [initial] [rendered]
chunk {styles} styles.bundle.js, styles.bundle.js.map (styles) 11.3 kB {inline}
[initial] [rendered]
chunk {vendor} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.29 MB [initial]
[rendered]

webpack: Compiled successfully.
```

A browser tab/window should open, loading the resource [localhost port 4200 URL](#).

## Edit the app

Edit the app, using Visual Studio Code.

An easy edit, just to prove that you can do so, is to edit the `app.component.ts` file in the `src/app` folder. Change the text in the value of the “title” property.

After you save your changes, switch over to the browser window. It should show the new content. Behind the scenes, the Terminal process will regenerate the content, making changes where necessary to the deployment assets. As part of the process, the browser will refresh the visible content.

# Welcome to my dream app!



## Understanding app startup

The following image shows how a new generated Angular app starts up. It shows the code and markup connections.

## index.html

This is the app's entry point

```
<body>
  <app-root></app-root>
</body>
```




## app.component.ts

This code is for the app's first generated component

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'feb06v1';
}
```



## app.component.html

This markup is in the app's first generated component

```
<div>
  <!-- more content here -->
</div>
```

## Work through some documentation topics

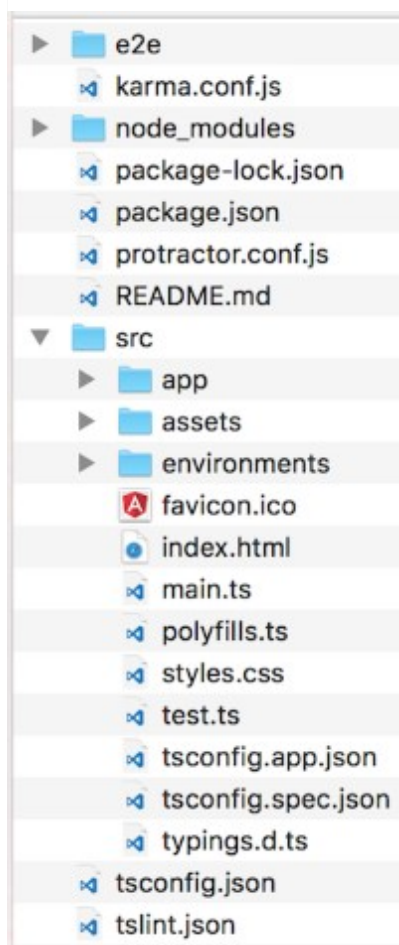
Return to the [GETTING STARTED](#) documentation topic.

In the instructions above, we have taken you through the first few screenfuls. Now it's time to read/scan the remaining content on this page, starting with [Step 4](#).

Take your time. While you are doing this, inspect the source code in your editor.

Here are some of the highlights that you should be looking for:

- The code for your app lives in the `src` folder
- The “ng new” generator creates a `src/app` folder, with code assets for the “app” component
- Later (but soon), we will use the Angular CLI to create another component in the `src` folder
- There are a number of files and folders that perform specific tasks for the app, and/or help you as the programmer



## What's next?

There are some documents that can help you understand more about the Angular way to build an app.

The official [Angular tutorial](#) is a recommended skim/read, and you may or may not want to code along with it.

*We recommend that you develop locally on your own computer (and not do the live online process).*

The other is from the Visual Studio Code documentation set, titled [Using Angular in VS Code](#). It's a reasonably short document. One of its very useful topics is the how-to for debugging Angular in VS Code.

Happy coding!

---

© 2020 - Seneca School of ICT