New project: ng new projectName

New component:

ng generate component <component-name> Or

ng g c <component-name>

ng g c path/<component-name>

ng g c –spec false | to prevent creation of test 🡪 Old version to do this

ng g c –skipTests true | to prevent creation of test 🡪 New version to do this

Bindable Properties and Events

How do you know to which Properties or Events of HTML Elements you may bind? You can basically bind to all Properties and Events - a good idea is to console.log()  the element you're interested in to see which properties and events it offers.

**Important**: For events, you don't bind to onclick but only to click (=> (click)).

The MDN (Mozilla Developer Network) offers nice lists of all properties and events of the element you're interested in. Googling for YOUR\_ELEMENT properties  or YOUR\_ELEMENT events  should yield nice results.

**Important:** FormsModule is Required for Two-Way-Binding!

Important: For Two-Way-Binding (covered in the next lecture) to work, you need to enable the ngModel  directive. This is done by adding the FormsModule  to the imports[]  array in the AppModule.

You then also need to add the import from @angular/forms  in the app.module.ts file:

import { FormsModule } from '@angular/forms';

**Setting bootstrap:**

**npm i bootstrap@3**

**When installing bootstrap 3, the messages needs to be ignored.**

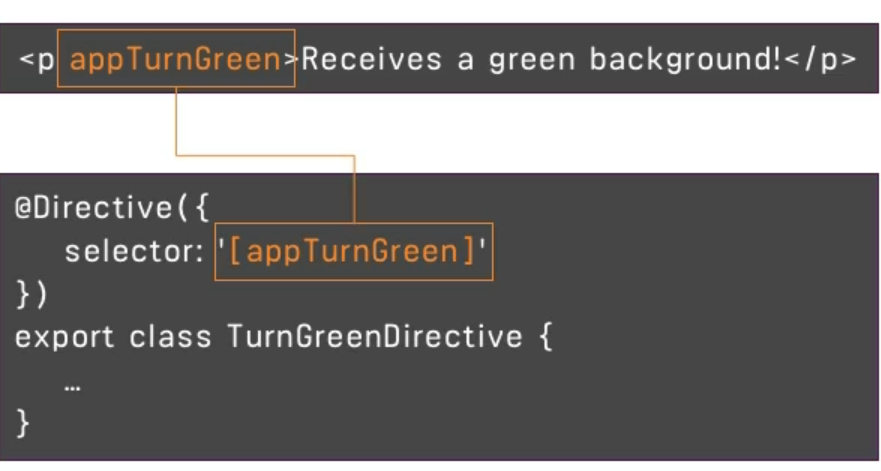
Add the **Bootstrap** path to the **build** section of your **angular.json**:

1. "build": {
2. ...
3. "styles": [
4. "node\_modules/bootstrap/dist/css/bootstrap.min.css",
5. "src/styles.css"
6. ],
7. ...

What are directives?

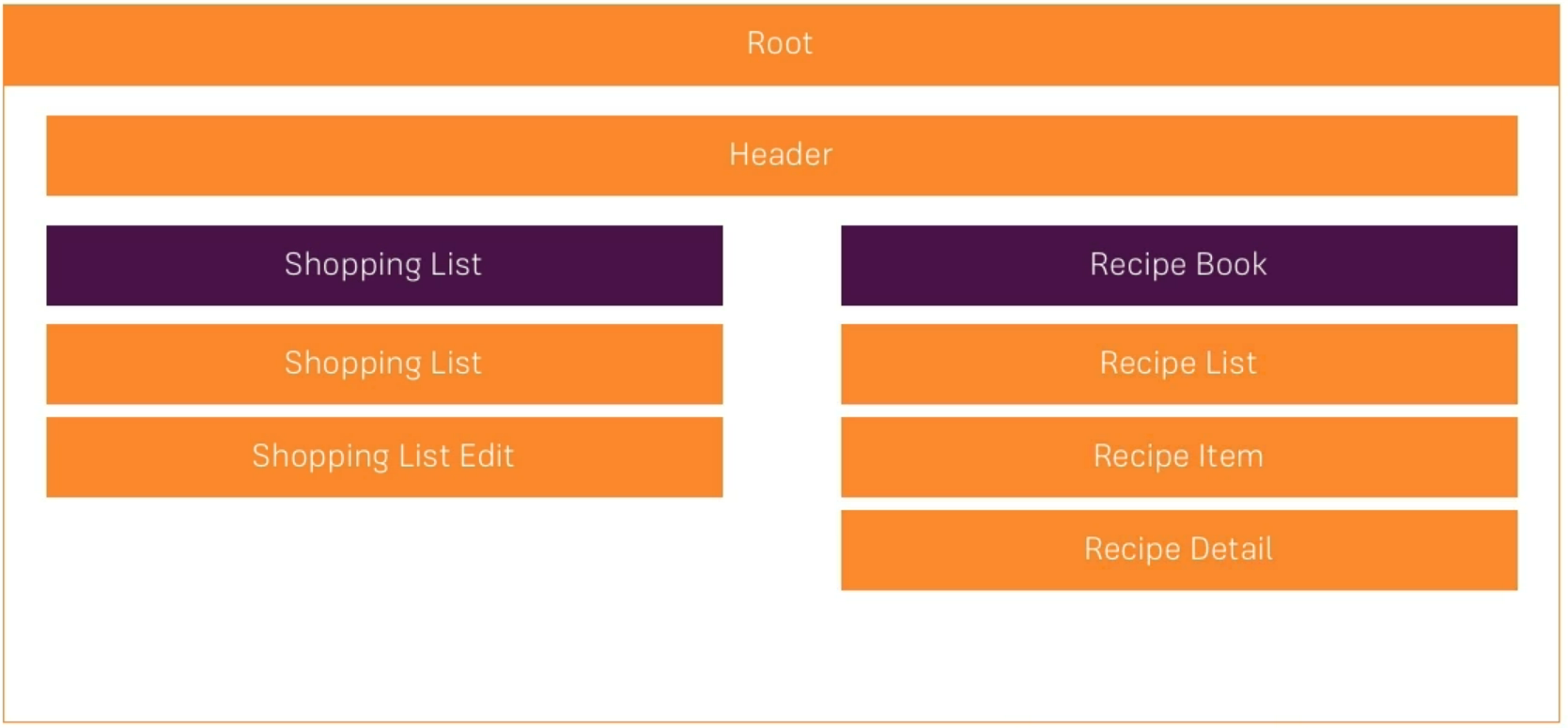
Directives are instructions in the DOM 🡪 Like components

Examples:



Section 3: Course Project – The basics

Planning the App







Creating a New App Correctly

**MUST READ**

In the next lecture, we set up the course project.

Make sure, you do create that app by also adding the --no-strict flag to the ng new command - otherwise you will run into issues later on (we'll still dive into that "Strict Mode" later in the course of course, no worries)!

We'll also install the Bootstrap CSS Framework and, in this course,, we use version 3 of the framework. Install it via npm install --save bootstrap@3  => The @3  is important!

Additionally, when using a project created with Angular CLI 6+ (check via ng v ), you'll have an angular.json  file instead of an .angular-cli.json  file. In that file, you still need to add Bootstrap to the styles[]  array as shown in the next video, but the path should be node\_modules/bootstrap/dist/css/bootstrap.min.css , **NOT** ../node\_modules/bootstrap/dist/css/bootstrap.min.css . The leading ../  must not be included.

Also see this lecture - I do show the complete setup process there: <https://www.udemy.com/the-complete-guide-to-angular-2/learn/v4/t/lecture/6655614/>

If you're facing any problems, please have a look at this very thorough thread by Jost: <https://www.udemy.com/course/the-complete-guide-to-angular-2/learn/lecture/17862130#questions/10444944>

**Important notes:**

2- Notes

1.

The course code works with **Bootstrap 3**, not with Bootstrap 4. Since in the meantime Bootstrap 4 was released, we have to declare explicitly which version we want to be installed.

So please write **npm i bootstrap@3** explicitly, not just **npm i bootstrap**.

(<https://www.udemy.com/the-complete-guide-to-angular-2/learn/v4/t/lecture/9118764>)

You might have a look at my fifth post below why installing version 3 - and not 4 - is important.

If you previously installed Bootstrap 4 accidentally, just run **npm i bootstrap@3** and restart the server. The wrong version will be uninstalled automatically.

You might notice that Max uses Bootstrap 3.3.7, and nowadays we get version 3.4.1 when running **npm i bootstrap@3**.But that's okay. After the video was recorded, a new version was released to fix a security issue (which would not be relevant for our app, btw.).

2.

In former times it was also possible to write the **node\_modules** path with **../** at the beginning. But that's no longer possible, like pointed out by Max in this slide:

<https://www.udemy.com/course/the-complete-guide-to-angular-2/learn/lecture/9118764>

After you have applied changes, don't forget step C from above (**restarting the server**)!

3.

The **angular.json** file is a replacement for the **.angular-cli.json** file found in older Angular versions.

4.

Make sure to add the **node\_modules** path not in the **test** section of **angular.json**, but in the **build** section.

5.

(a) If you don't see Bootstrap added in the **node\_modules** folder of your project explorer in **VSCode**'s left side bar, please hover over the project explorer's top bar, and then click on the appearing **refresh** button.

(b) If you don't see the Bootstrap styles in the browser's **Elements** tab as a separate <style> tag - no worries: Since Angular 11 the Bootstrap styles are shown there as part of the linked **styles.css** file (just right-click on the **styles.css** link to open this file in a new tab). But that's an implementation detail which doesn't matter.

6.

The **--save** flag used by Max is no longer required for **npm** installations, but you can still use it to be more explicit about the fact that the package is added by default to the **dependencies** in the project's **package.json**.

7.

On purpose in the course project only **bootstrap's CSS** is used, and **not bootstrap's JS** functionality (**JQuery**, **Popper** etc.).

In this way a **very** important principle is respected: In general it's **highly** recommended in an Angular app that Angular should be the only one accessing the DOM directly!

This statement applies regardless of the Bootstrap version!

The dropdowns and a responsive navbar will be implemented throughout this course, using **Angular's** techniques in connection with **Bootstrap's** CSS only.

In real life you don't need to create all UI elements on your own, of course. Please have a look at "Angularish" solutions like **Angular Material**, **NG Bootstrap**, **NGX-Bootstrap**, **PrimeNG** etc..

Emmet Plugin for bootstrap

Alternative Non-Collapsable Navigation Bar

The way we added it, the Navbar will collapse on smaller screens. Since we didn't implement a Hamburger menu, that means that there's no way of accessing our links on smaller screens.

You can either add such a menu on your own (see below), or you replace collapse navbar-collapse  with just navbar-default.

Adding a Hamburger Menu:

Alternatively, if you want to make the navigation bar responsive, please replace these lines in header.component.html:

1. <div class="navbar-header">
2. <a routerLink="/" class="navbar-brand">Recipe Book</a>
3. </div>
4. <div class="collapse navbar-collapse">

with these lines:

1. <div class="navbar-header">
2. <button type="button" class="navbar-toggle" (click)="collapsed = !collapsed">
3. <span class="icon-bar" \*ngFor="let iconBar of [1, 2, 3]"></span>
4. </button>
5. <a routerLink="/" class="navbar-brand">Recipe Book</a>
6. </div>
7. <div class="navbar-collapse" [class.collapse]="collapsed" (window:resize)="collapsed = true">

and add this line to header.component.ts:

1. collapsed = true;

--spec false