CLI Deep Dive & Troubleshooting

Section 1, Lecture 4

In the next lecture, we're going to build our first little app!

If the CLI prompts you to**answer some questions**(some versions do that), you can simply hit **ENTER** for all questions. This will accept the default settings which are fine for this course.

The CLI generates a different welcome screen than you're going to see in my video though. No worries, you'll still be able to follow along without issues! Just make sure to code along **so that your code equals mine** - Angular itself didn't change a bit :)

Depending on the CLI version you're using, you might also need to add the FormsModule  to the imports[]  array in your app.module.ts  file (add it if you don't see it there). You might not fully understand what that all means but we're going to cover that in this course, no worries.

If you don't have FormsModule  in imports[]  in AppModule , please do add it and also add an import at the top of that file: import { FormsModule } from '@angular/forms';

-----

If you want to **dive deeper into the CLI** and learn more about its usage, have a look at its official **documentation**: <https://github.com/angular/angular-cli/wiki>

**You encountered issues during the installation of the CLI or setup of a new Angular project?**

A lot of problems are solved by making sure you're using the latest version of NodeJS, npm and the CLI itself.

**Updating NodeJS:**

Go to nodejs.org and download the latest version - uninstall (all) installed versions on your machine first.

**Updating npm:**

Run [sudo] npm install -g npm  (sudo  is only required on Mac/ Linux)

**Updating the CLI**

[sudo] npm uninstall -g angular-cli @angular/cli

npm cache clean

[sudo] npm install -g @angular/cli

**Here are some common issues & solutions:**

1. **Creation of a new project takes forever (longer than 3 minutes)**  
   That happens on Windows from time to time => Try running the command line as administrator
2. **You get an EADDR error (Address already in use)**  
   You might already have another ng serve process running - make sure to quit that or use ng serve --port ANOTHERPORT  to serve your project on a new port
3. **My changes are not reflected in the browser (App is not compiling)**  
   Check if the window running ng serve  displays an error. If that's not the case, make sure you're using the latest CLI version and try restarting your CLI



**Commands**

1. npm install -g @angular/cli
2. ng new my-first-app
3. ng serve



1. npm install –save bootstrap@3
2. ng generate component servers

ng g c servers

If you want to output something in your template print some text to it use string interpolation, if you want to change some property be that of an angular element or as you will later learn of a directive or a component.

No {{}} braces for property binding

Bindable Properties and Events

**Section 2, Lecture 27**

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!**

**Section 2, Lecture 29**

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';

**Directives**

star is required because ng-if is a structural directive which means it changes the structure of our DOM. It either adds this element or it doesn’t add it.

Unlike structural directives, attribute directives don’t add or remove elements. They only change the elements they are placed on. They look like normal HTML attributes without a star basically.

**Installing Bootstrap Correctly**

**Section 3, Lecture 41**

In the next lecture, we set up the course project. For that, we'll install the Bootstrap CSS Framework.

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/>

**Alternative Non-Collapsable Navigation Bar**

**Section 3, Lecture 46**

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, or you replace collapse navbar-collapse with just navbar-default