Setting Up A Material Design Angular Project

1. Bootstrap an angular app using angular-cli (command line interface)\
   1. Tips – make sure node and npm is installed
   2. Tips – install angular-cli
   3. Tips – check versions by typing ng – version (angular-cli)
2. Create a new project (sass enabled)
   1. Ng new YourProjectNameHere --style=scss (or without)
   2. Note – it will take some time to load this project check the npm folder to see if it loads up about 800+ libraries
3. Cd into the project > cd YourProjectNameHere

USE MATERIAL DESIGN :

Install Angular Material, Angular animations, and Hammer.js :

* npm install --save @angular/material @angular/animations @angular/cdk
* npm install --save hammerjs (Hammer.js is an optional dependency and helps with touch support for a few of the components)
* if you decide to use Hammer.js, and given that you’ve started your project with the Angular CLI, modify your **angular-cli.json** file to add the Hammer.js library. Look for the Json “scripts” array and add the following path for hammer.js:  
  "scripts": [ "../node\_modules/hammerjs/hammer.min.js"],

Prior to Angular Material 2 Beta 3, there was a global MaterialModule that could be imported in the app module to make the components available. **MaterialModule has therefore been deprecated** in favor of defining **a project-specific custom material module** where you import and export only the needed components.

* Create a custom material module for project > ng g class material.module
* Change export value to MaterialModule – should look like export class MaterialModule {}
* Add Angular Material to your app module app.module.ts
  + import { MaterialModule } from './material.module';

imports: [

BrowserModule,

FormsModule,

HttpModule,

MaterialModule,

BrowserAnimationsModule

],

ADD A THEME TO THE TOOLBAR:

* add an import in the style.scss file :

@import '../node\_modules/@angular/material/prebuilt-themes/deeppurple-amber.css';

* use the theme in html toolbar like : <mat-toolbar color="primary">  
  \*see project for creating custom themes

MAKING THE LAYOUT RESPONSIVE: using flex-layout

* add the flex-layout angular dependency : npm install –save @angular/flex-layout
* add the reference to the app.module.ts : import { FlexLayoutModule } from '@angular/flex-layout';
* Add some refernces in the app.component html to see if it works
  + - <div class="content" fxLayout="row" fxLayout.xs="column" fxFlexFill >
    - <div fxFlex="25" class="sec1" fxFlex.xs="50">
    - first-section
    - </div>
    - <div fxFlex="75" class="sec3" fxFlex.xs="50">
    - third-section
    - </div>
    - </div>

1. Build + Run to make sure it’s a clean load > ng build >ng serve

**Outdated Version error :**

* 1. note: if you get a bunch of outdated file errors then you may need to update the version of angular-cli by doing the following WITHIN THE PROJECT:
     1. rm -rf node\_modules dist
     2. npm install --save-dev @angular/cli@latest
     3. npm install
  2. note : check the @angular folder after node\_modules reinstalls for material folder
  3. note: rebuild and re-serve to ensure the app still works.> ng build ng serve
  4. open the text editior to app.module.ts(src/app/) file add the following>