

Exercise (Instructions): Configuring your Angular Application

Objectives and Outcomes

In this exercise we will set up our project to use Angular Material and Angular Flex Layout. We will then introduce our first Angular Material component into our application. At the end of this exercise you will be able to:

- Configure your Angular project to use Angular Material and Flex Layout.
- Start using Material components in your application.

Configure your Angular Project to use Angular Material

Note: This course is designed with Angular Material Beta.3. Before you proceed forward, you may wish to read the detailed information posted in

<https://www.coursera.org/learn/angular/discussions/all/threads/4yxVk7DXEee0mQrUfDuicA> where I have clearly explained about dealing with the newer Beta versions of Angular Material (up to Beta.12). I would strongly suggest that to proceed ahead with the course with minimal disruption, please install the Beta.8 version of Angular Material. With this installation, the course instructions will still work as given.

- To configure your project to use Angular material, type the following at the prompt to install Angular Material, Angular Animations and HammerJS:

```
1 npm install @angular/material@2.0.0-beta.8 --save
2 npm install @angular/cdk@2.0.0-beta.8 --save
3 npm install --save @angular/animations
4 npm install --save hammerjs
```

Configure to use Material Design Icons

- Next, include the following into the <head> of *index.html* to make use of Material Design icons:

```
1 <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel
  ="stylesheet">
```

Configure your Angular Project to use Flex Layout

- Next, install Angular Flex Layout as follows:

```
1 npm install --save @angular/flex-layout@latest |
```

Updating AppModule

- Then, you need to import the Angular Animations Module, Angular Material Module, Flex Layout Module and hammerjs into your root module (src/app/app.module.ts) as follows:

```
1  . . .
2
3  import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
4  import { MaterialModule } from '@angular/material';
5  import { FlexLayoutModule } from '@angular/flex-layout';
6
7  . . .
8
9  import 'hammerjs';
10
11  @NgModule({
12
13    . . .
14
15    imports: [
16
17      . . . ,
18
19      BrowserAnimationsModule,
20      MaterialModule,
21      FlexLayoutModule
22
23    ],
24
25    . . .
26
27  })
28
29  . . .
30  . . . |
```

Adding a Material Toolbar

- Open app.component.html and replace its contents with the following code:

```
1 <md-toolbar color="primary"> <span>Ristorante Con Fusion</span> </md-toolbar> |
```

Adding Styles

- Add the following styles to styles.scss file:

```
1 @import '~@angular/material/prebuilt-themes/deeppurple-amber.css';
2
3 // some basic resets
4
5 body {
6   padding: 0;
7   margin: 0;
8   font-family: Roboto, sans-serif;
9 }
10 |
```

- This will add a built-in Material theme to our application.
- Do a Git commit with the message "Configuring Angular"

Conclusions

In this exercise we learnt to use Angular Material and Flex Layout NgModules in our Angular application.

Mark as completed

