CodingTheSmartWay.com

≡

Material Design With Angular 2 Tutorial

▶

ANGULAR

POSTED BY **SEBASTIAN** — 11TH SEPTEMBER '16

Using Material Design In Angular 2

To get started with Angular 2 and Material Design just go through the following steps:

1) Start with a new project setup

By following the Angular 2 quickstart tutorial you can set up a new base project structure:

|-app

—|-app.component.ts

```
–|-app.module.ts
–|-main.ts
|-index.html
|-package.json
|-systemjs.config.js
|-tsconfig.json
|-typings.json
```

2) Introduce Material Design

Angular 2 Material Design Components are available as NPM packages. A full list of packages can be found at https://www.npmjs.com/~angular2-material.

A general introduction and overview of Google's Material Design can be found at https://material.google.com/.

Created and designed by Google, Material Design is a design language that combines the classic principles of successful design along with innovation and technology. Google's goal is to develop a system of design that allows for a unified user experience across all platform.

3) Install Angular 2 Material Design NPM Packages

Free ebook | Essentials - ECMAScript 6 and TypeScript - *Get Your Copy Now*

| Name | Email | DOWNLOAD (X) |

$ npm install —save @angular2-material/core @angular2-material/button @angular2-material/card @angular2-material/icon

Note: The core module is always required as a peer dependency of other components

4) Install HammerJS

We need to add the third-party library HammerJS to our
project. Hammer add support for touch gestures and is
used by the Icon Material components. HammerJS is
available as a NPM package, so we can install and add it
to the project dependencies by using the following
command:

$ npm install hammerjs —save

And the corresponding typings via:

$ typings install dt~hammerjs —save –global

5) SystemJS Configuration

Next, add the Angular 2 Material NPM packages to
systemjs.config.js by using the UMD (Universal Module
Definition) files:

```
map: {
// our app is within the app folder
app: 'app',
// angular bundles
'@angular/core': 'npm:@angular/core/bundles/c
```

```
'@angular/platform-browser-dynamic': 'npm:@ar
'@angular/http': 'npm:@angular/http/bundles/h
'@angular/router': 'npm:@angular/router/bundl
'@angular/forms': 'npm:@angular/forms/bundles
// other libraries
'rxjs': 'npm:rxjs',
'angular2-in-memory-web-api': 'npm:angular2-i
```

```
'@angular2-material/core': 'npm:@angular2-mat
'@angular2-material/card': 'npm:@angular2-mat
'@angular2-material/button': 'npm:@angular2-m
'@angular2-material/icon': 'npm:@angular2-mat
},
```

6) Import Material Modules and Providers in AppModule

We are only using one Angular 2 module in our application (*AppModule*). We need to make the Angular 2 Material elements available within this module. First add the corresponding import statements:

```
import {MdCardModule} from '@angular2-materia
import {MdButtonModule} from '@angular2-mater
import {MdIconModule} from '@angular2-materia
import {MdIconRegistry} from '@angular2-mater
```

Next the modules needs to be added to the array which is assigned to the imports property of the *@NgModule* decorator. *MDIconRegistry* needs to be added to the array which is assigned to the providers property:

```
@NgModule({
imports: [ BrowserModule, MdCardModule, MdBut
declarations: [ AppComponent ],
bootstrap: [ AppComponent ],
providers: [ MdIconRegistry ]
})
export class AppModule { }
```

7) Include Google Material Icon Font in index.html

By default Material Icons make us of the Google Material Icon Font, so we need to import the font in index.html:

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

8) Implement AppComponent template in app.component.html

We're implementing the component's template by making use of Material Card, Buttons and Icons:

```html
<div class="app-content">
<md-card>
<button md-button>FLAT</button>
<button md-raised-button>RAISED</button>
</md-card>
<md-card>
<button md-icon-button>
<md-icon class="md-24">favorite</md-icon>
</button>
<button md-fab>
<md-icon class="md-24">add</md-icon>
</button>
<button md-mini-fab>
<md-icon class="md-24">add</md-icon>
</button>
</md-card>
<md-card>
<button md-raised-button color="primary">PRIM
```

```
<button md-raised-button color="accent">ACCEN
<button md-raised-button color="warn">WARN</t
</md-card>
<md-card>
<button md-button disabled>OFF</button>
<button md-raised-button disabled>OFF</buttor
<button md-mini-fab disabled><md-icon>check</
</md-card>
</div>
```

Include the external template file in the component
decorator:

```
import { Component } from '@angular/core';
@Component({
selector: 'my-app',
templateUrl: './app/app.component.html'
})
export class AppComponent { }
```
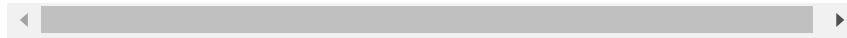
9) Add some styling

In addition we can add some CSS styling. Add the
styleURLs property to *@Component* and point to the
external styling file *app.component.css*:

```
import { Component } from '@angular/core';
@Component({
selector: 'my-app',
templateUrl: './app/app.component.html',
```

```
styleUrls: ['./app/app.component.css']
})
export class AppComponent { }
```

◀ ▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐ ▶

Create the CSS file in the app folder and insert the
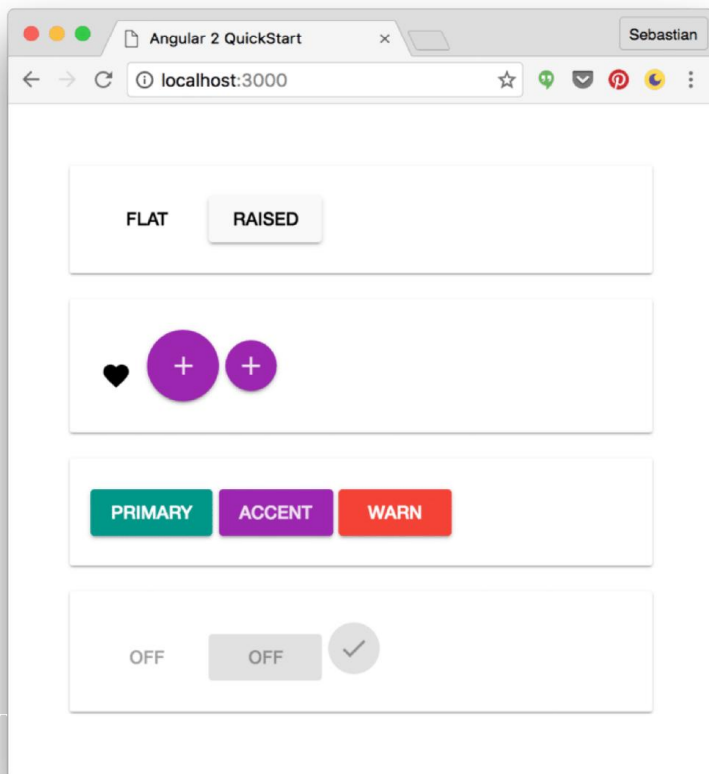following content:

```
.app-content {
padding: 20px;
}
.app-content md-card {
margin: 20px;
}
```

10) Final Result

Having finished the implementation we can start up the
application by using the following command:

$ npm start

and view the final result in the browser:

Essentials - ECMAScript 6 and TypeScript

Comprehensive introduction! **Get your free copy today!**

| Name | Email | Download |

PREVIOUS POST

ANGULAR

Getting Started With Angular 2 Modules (Using The @NgModule Decorator)

@s_eschweiler
Sebastian

Using and writing about best practices and latest technologies in web design & development is my passion.

ℹ There are currently no comments.

Add your comment here...

Name (Required)   **Email (Required)**   **Website URL**

CAPTCHA Code*

POST COMMENT

Copyright © CodingTheSmartWay.com. 2016 • All rights reserved.

Imprint