

Getting Started with Angular Material

Deserae Dias

...Material

2:15 PM - 3:15 PM

TDD (Test Driven Development)

Yogesh Edekar



Getting started with Angular ...

Deserae Dias

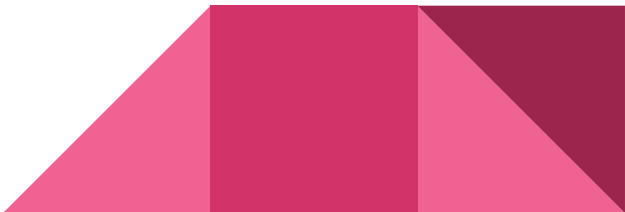


Leveraging Go to Build a FaaS ...

Brian Downs



Available



About

This presentation will provide a brief overview to Angular Material, an Angular component library for styling based off of the Material design language created by Google.

This talk assumes little to no knowledge of either Angular or Angular Material



Table of Contents

- Intro
- Background
- Features
- Examples
- Resources
- Outro



About me

- Software Developer
- RockIT Bootcamp Alum
- Using Angular Material since 2018

Twitter: @DeseraeDias

GitHub: Deserae





Background

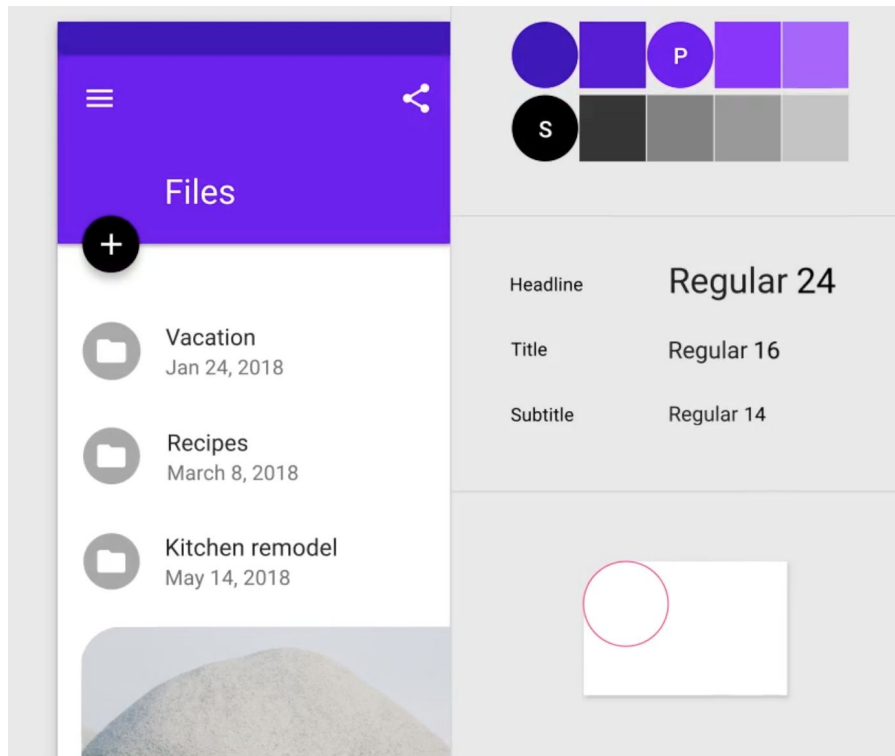
Angular

- Web Application Framework for Single Page Applications
- Uses Typescript (superscript of Javascript)
- Component based structure
- Open source & led by Google
- AngularJS released 2010. Angular2 released 2016
- **AngularJS vs Angular2+**
 - 2 was a complete rewrite
 - *Protip* Specify when Googling
 - Angular 6 released Sept. 2018



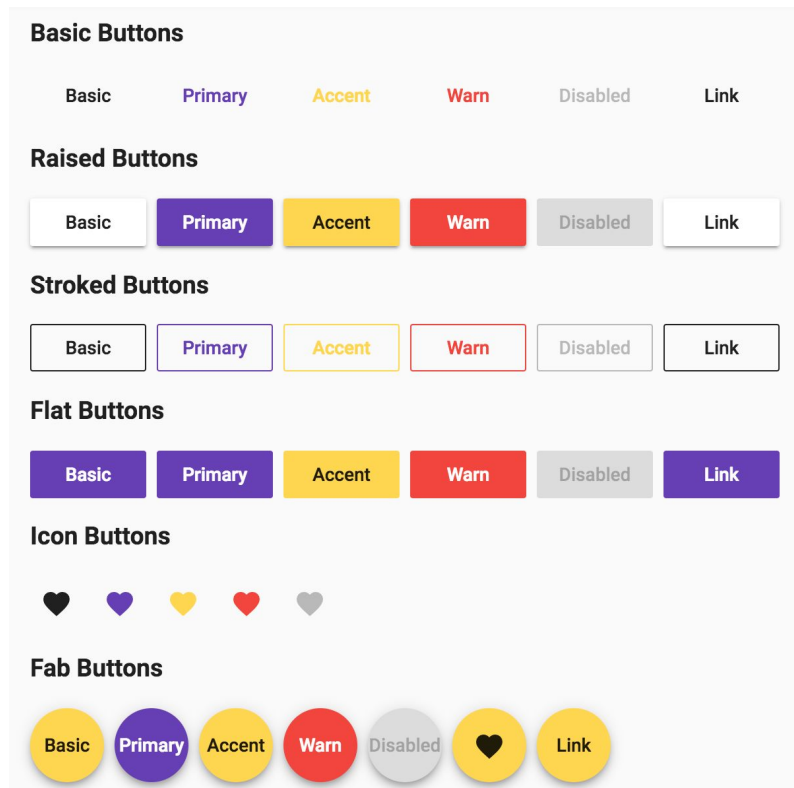
Material Design

- Design language created by Google in 2014
- Notable for its cards, semi flat-design, use of shadow, and responsive animation
- Has been implemented in many Google products including Gmail, Drive, and Youtube.
- Amazing documentation!



Angular Material

- Library of UI components for Angular
- Follow Material guidelines to create a consistent look
 - Customizable within the limits of Material guidelines
- Create CSS “Themes”
- Available for both AngularJS and Angular2+





NORMAL

NORMAL



PRESSED

PRESSED



Input tex|

Permissions

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed
do eiusmod tempor incididunt...

BUTTON

BUTTON



Benefits

- Built for Angular
- Cross browser support
- ARIA / Screen reader support
- Consistent design
- Save time on design decisions
- You suck at CSS



Features

Themes

- Default themes or make your own
- Needs to be imported/linked to your styles.css
- Can have 1 global theme or multiple for different components



```
@import '~@angular/material/theming';
// Plus imports for other components in your app.

// Include the common styles for Angular Material. We include this here so that you only
// have to load a single css file for Angular Material in your app.
// Be sure that you only ever include this mixin once!
@include mat-core();

// Define the palettes for your theme using the Material Design palettes available in palette.scss
// (imported above). For each palette, you can optionally specify a default, lighter, and darker
// hue. Available color palettes: https://material.io/design/color/
$candy-app-primary: mat-palette($mat-indigo);
$candy-app-accent:  mat-palette($mat-pink, A200, A100, A400);

// The warn palette is optional (defaults to red).
$candy-app-warn:    mat-palette($mat-red);

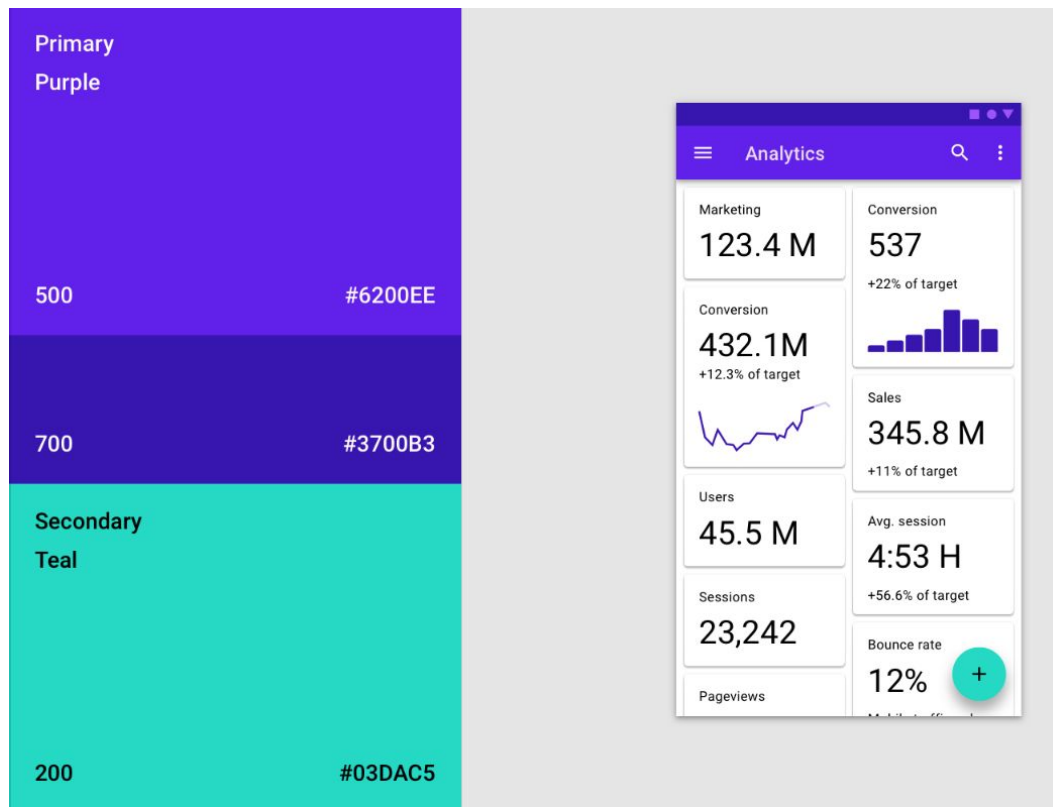
// Create the theme object (a Sass map containing all of the palettes).
$candy-app-theme: mat-light-theme($candy-app-primary, $candy-app-accent, $candy-app-warn);

// Include theme styles for core and each component used in your app.
// Alternatively, you can import and @include the theme mixins for each component
// that you are using.
@include angular-material-theme($candy-app-theme);
```

Primary	Primary Variant	Secondary	Secondary Variant
1 #6200EE	2 #3700B3	3 #03DAC6	4 #018786

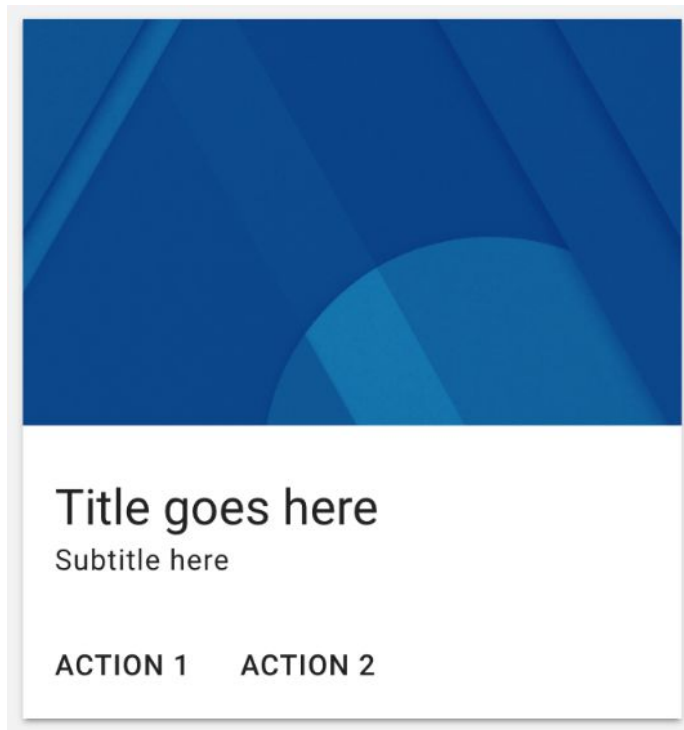
Background	Surface	Error
5 #FFFFFF	6 #FFFFFF	7 #C51162

On Primary	On Secondary	
8 #FFFFFF	9 #000000	
On Background	On Surface	On Error
10 #000000	11 #000000	12 #FFFFFF



Cards

- Cards are a major design feature of the Material design
- “Content container for text, photos, and actions in the context of a single subject”
- You’ve seen them



```
<mat-card class="example-card">
  <mat-card-header>
    <div mat-card-avatar class="example-header-image"></div>
    <mat-card-title>Shiba Inu</mat-card-title>
    <mat-card-subtitle>Dog Breed</mat-card-subtitle>
  </mat-card-header>
  
  <mat-card-content>
    <p>
      The Shiba Inu is the smallest of the six original and
      distinct spitz breeds of dog from Japan.
      A small, agile dog that copes very well with mountainous
      terrain, the Shiba Inu was originally
      bred for hunting.
    </p>
  </mat-card-content>
  <mat-card-actions>
    <button mat-button>LIKE</button>
    <button mat-button>SHARE</button>
  </mat-card-actions>
</mat-card>
```



Shiba Inu

Dog Breed



The Shiba Inu is the smallest of the six original and distinct spitz breeds of dog from Japan. A small, agile dog that copes very well with mountainous terrain, the Shiba Inu was originally bred for hunting.

LIKE

SHARE



Cafe Badilico

★★★★★ 4.5 (413)

\$ • Italian, Cafe

Small plates, salads & sandwiches an intimate setting with 12 indoor seats plus patio seating.

Tonight's availability

5:30PM

7:30PM

8:00PM

9:00PM

[RESERVE](#)

Word of the Day

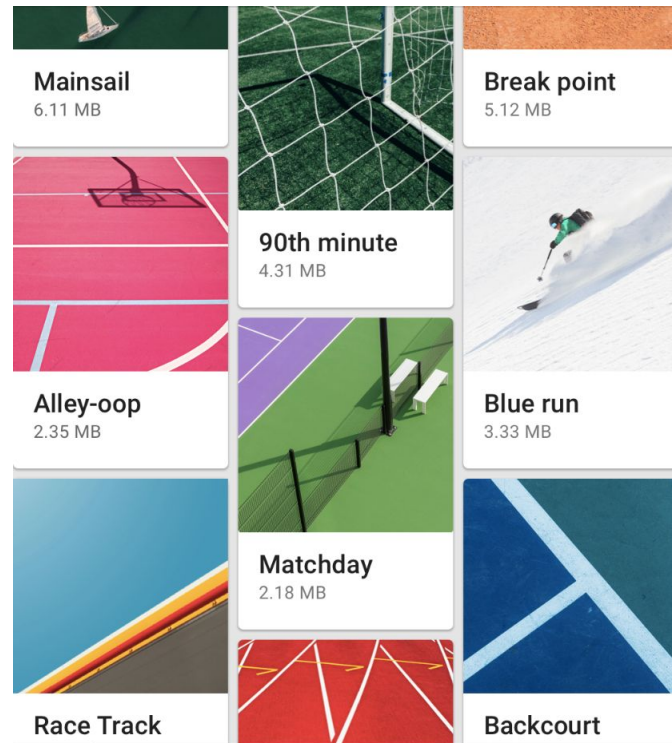
be•nev•o•lent

adjective

well meaning and kindly.
“a benevolent smile”

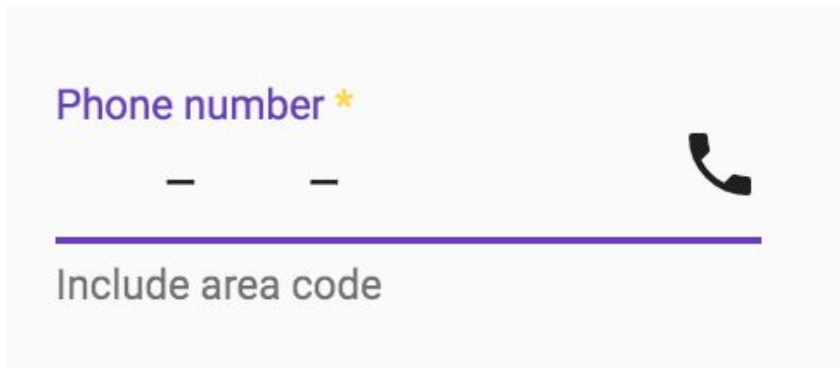
synonyms: [kind](#), [kindly](#), [kindhearted](#), [big-hearted](#),
[good-natured](#), [good](#), [benign](#), [compassionate](#),
[caring](#), [altruistic](#), [humanitarian](#), [philanthropic](#);

[LEARN MORE](#)



Form fields

- You're going to need them!
- Pain point for users (and developers)
- UI validation
- Multiple appearances to choose from



Phone number *

— —

Include area code

A black telephone handset icon is located on the right side of the form field.

The image shows a form field for a phone number. It has a label "Phone number *" in purple, a placeholder with two dashes "— —", and a hint text "Include area code" below a horizontal line. A black telephone handset icon is on the right.

```

<p>
  <mat-form-field appearance="legacy">
    <mat-label>Legacy form field</mat-label>
    <input matInput placeholder="Placeholder">
    <mat-icon matSuffix>sentiment_very_satisfied</mat-icon>
    <mat-hint>Hint</mat-hint>
  </mat-form-field>
</p>
<p>
  <mat-form-field appearance="standard">
    <mat-label>Standard form field</mat-label>
    <input matInput placeholder="Placeholder">
    <mat-icon matSuffix>sentiment_very_satisfied</mat-icon>
    <mat-hint>Hint</mat-hint>
  </mat-form-field>
</p>
<p>
  <mat-form-field appearance="fill">
    <mat-label>Fill form field</mat-label>
    <input matInput placeholder="Placeholder">
    <mat-icon matSuffix>sentiment_very_satisfied</mat-icon>
    <mat-hint>Hint</mat-hint>
  </mat-form-field>
</p>
<p>
  <mat-form-field appearance="outline">
    <mat-label>Outline form field</mat-label>
    <input matInput placeholder="Placeholder">
    <mat-icon matSuffix>sentiment_very_satisfied</mat-icon>
    <mat-hint>Hint</mat-hint>
  </mat-form-field>
</p>

```

Legacy form field

Placeholder



Hint

Standard form field



Hint

Fill form field



Hint

Outline form field



Hint

Email

Errors appear instantly!

Enter your email *

You must enter a value

```
<form class="example-form">
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Email" [formControl]="
      emailFormControl"
      [errorStateMatcher]="matcher">
    <mat-hint>Errors appear instantly!</mat-hint>
    <mat-error *ngIf="emailFormControl.hasError('email') &&
      !emailFormControl.hasError('required')">
      Please enter a valid email address
    </mat-error>
    <mat-error *ngIf="emailFormControl.hasError('required')">
      Email is <strong>required</strong>
    </mat-error>
  </mat-form-field>
</form>
```



```

import {Component} from '@angular/core';
import {FormControl, FormGroupDirective, NgForm, Validators} from '@angular/forms';
import {ErrorStateMatcher} from '@angular/material/core';

/** Error when invalid control is dirty, touched, or submitted. */
export class MyErrorStateMatcher implements ErrorStateMatcher {
  isErrorState(control: FormControl | null, form: FormGroupDirective | NgForm | null): boolean {
    const isSubmitted = form && form.submitted;
    return !(control && control.invalid && (control.dirty || control.touched || isSubmitted));
  }
}

/** @title Input with a custom ErrorStateMatcher */
@Component({
  selector: 'input-error-state-matcher-example',
  templateUrl: './input-error-state-matcher-example.html',
  styleUrls: ['./input-error-state-matcher-example.css'],
})
export class InputErrorStateMatcherExample {
  emailFormControl = new FormControl('', [
    Validators.required,
    Validators.email,
  ]);

  matcher = new MyErrorStateMatcher();
}>

```

Getting Started

1. Created an Angular app
2. `npm or yarn install: @angular/material @angular/cdk @angular/animations`
3. Configure animations
4. Import component modules: `imports: [BrowserAnimationsModule]`
5. Theme is required! `@import '~@angular/material/ prebuilt-themes/ deeppurple-amber.css';`
6. *Optional: Some components require HammerJS installed to work
7. *Optional: Install Material Icons



Examples

Resources

Material: <https://material.io/>

Angular Material: <https://material.angular.io/>

On themes:

<https://medium.com/@tomastrajan/the-complete-guide-to-angular-material-themes-4d165a9d24d1>

Angular Material IO 2018 talks:

https://youtu.be/3VUMI_l-_fl

<https://youtu.be/hA0hrpR-o8U>



Resources - Demos

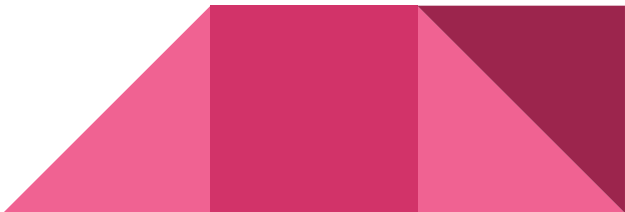
Material <https://material.io/>


Form <https://demo-071.codingthesmartway.com/create>

Basil <https://material.io/design/material-studies/basil.html>

Shrine <https://material.io/design/material-studies/shrine.html>

PokeDex <https://ng-pokedex.firebaseio.com/pokemon>



The background is a solid pink color. In the top right corner, there is a decorative pattern of overlapping triangles in various shades of pink and magenta, creating a geometric, stepped effect.

{{ Insert your
favorite cat meme
here }}



Questions?