Angular
~~Angular 2~~ Deep Dive

By Chris McKnight

About Me

About Me

- Graduated from LSU in Computer Science with a concentration in Mathematics
- Background in C++, PHP, and JavaScript

Overview

Many pieces of the Angular ecosystem to cover!

- Angular Cli (@angular/cli on npm)
- Angular: Directives, Services, Modules, and Pipes
- RxJS
- The HTTP service

Angular

> *Angular is a development platform for building mobile and desktop web applications.*

One framework.

Mobile & Desktop

Decorators

> *A function that adds metadata to a class, its members (properties, methods) and function arguments.*
>
> *Decorators are an experimental (stage 2), JavaScript language feature. TypeScript adds support for decorators.*

Also known as an `annotation`

Decorator example

```
import { deprecate } from 'core-decorators';
import { compact } from 'lodash';

class Person {
  ...
  // deprecate the name method using a decorator
  @deprecate('Use the fullName method instead')
  get name() {
    return compact([this.first_name, this.last_name]).join(' ');
  }

  get fullName() {
    ...
  }
}
```

Directives

An Angular class responsible for creating, reshaping, and interacting with HTML elements in the browser DOM.

The directive is Angular's most fundamental feature.

Three types of Directives

- Attribute directives
- Structural directives
- Components

Components

- Fundamental building block to Angular
- Handles exposing data to view and handling user interaction
- A directive containing a template

Components

```
import { Component } from '@angular/core';

@Component({
  selector: 'home',
  styles: [`
  h1 { color: rebeccapurple; }
  `],
  template: `
  <h1>{{title}}</h1>
  `
})
export default class HomeComponent {
  public title: string = 'Hello World';
}
```

Attribute Directives

A category of directive that can listen to and modify the behavior of other HTML elements, attributes, properties, and components. They are usually represented as HTML attributes, hence the name.

Common attribute directives:

- ngClass
- ngStyle

Structural Directives

A category of directive that can shape or reshape HTML layout, typically by adding and removing elements in the DOM.

- ngIf
- ngFor
- ngSwitch

Container components

a.k.a. ”smart” components

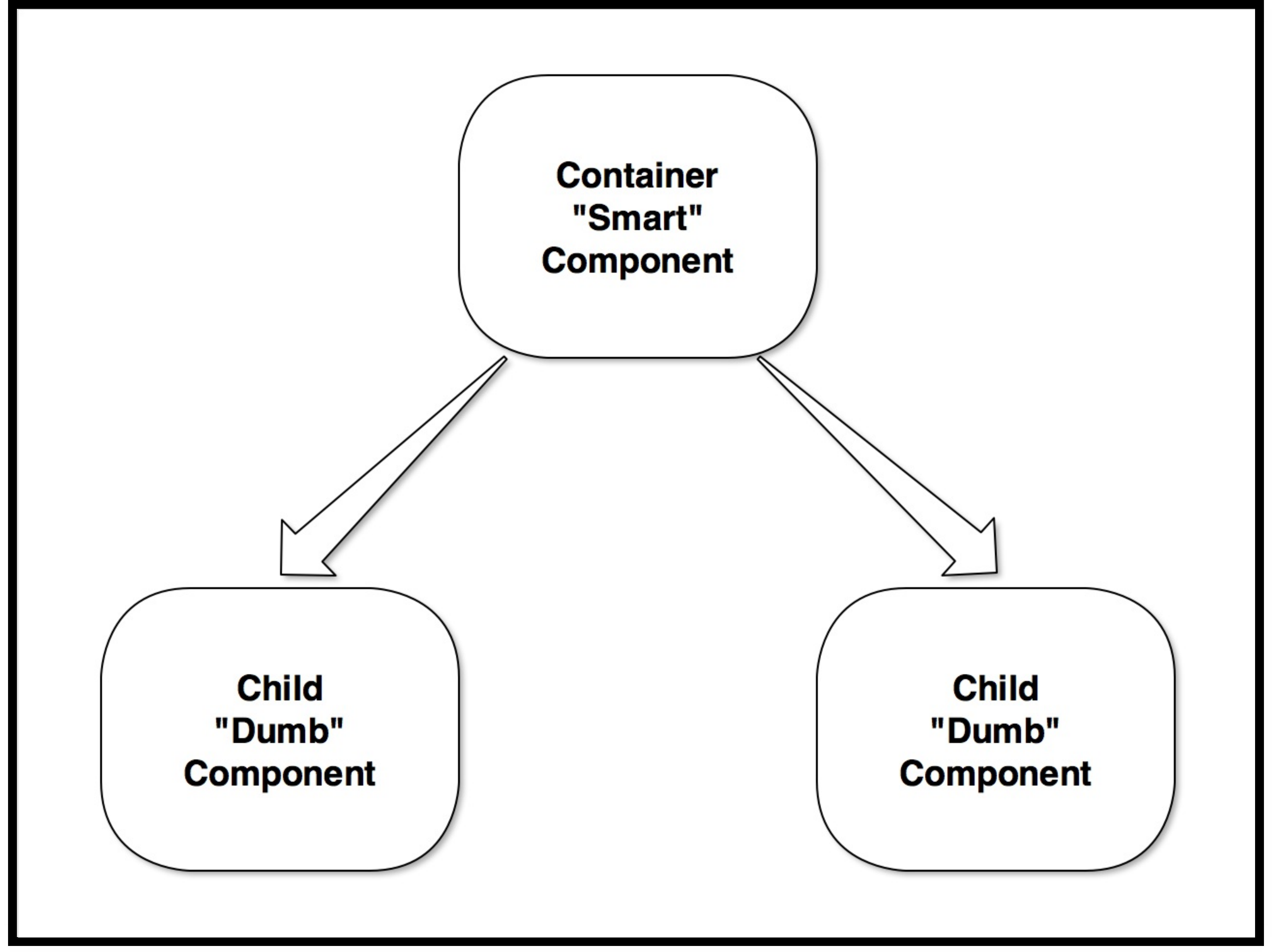
- Root level, routable components
- Access application state from ngrx, services, etc
- Handle events emitted from child components within the same view tree

Presentational Components

a.k.a. ”dumb” components

- Small
- Focused
- Reusable
- State down, Events up

Container & presentational components



Component lifecycle

Common lifecycle events

- ngOnInit
- ngAfterContentInit
- ngAfterViewInit
- ngOnDestroy

Pipes

An Angular pipe is a function that transforms input values to output values for display in a view.

```
...

@Component({
  selector: 'my-component',
  template: `
  <!-- Today is Apr 21, 2017 -->
  <span>Today is {{ currentDate | date }}</span>
  `
})
export class MyComponent {
  public currentDate = new Date();
}
```

Services

```
import { Injectable } from '@angular/core';

@Injectable()
export default class DataService {
  public title: string = 'My Application';
}
```

Modules

Helps you organize an application into cohesive blocks of functionality. An Angular module identifies the components, directives, and pipes that the application uses along with the list of external Angular modules that the application needs, such as FormsModule.

Module Example

```
import { NgModule, CommonModule } from '@angular/core';
import { HomeComponent } from './home.component';
import { HomeService } from './home.service';
export default class HomeModule {
  declarations: [
    HomeComponent
  ],
  imports: [CommonModule],
  exports: [HomeComponent],
  providers: [HomeService]
}
```

Introduction to RxJS

> *RxJS is a library for reactive programming using Observables, to make it easier to compose asynchronous or callback-based code.*

- Observable
- Subscription

Observables

> *The Observable object represents a push based collection.*

Observables are used extensively in Angular (`Http` service)

Subscription

> *A Subscription is an object that represents a disposable resource, usually the execution of an Observable.*

Subscription objects have a single method!

RxJS Example

```
const source = Rx.Observable.timer(200, 100)
    .take(3);

const subscription = source.subscribe(
    function (x) {
        console.log('Next: ' + x);
    },
    function (err) {
        console.log('Error: ' + err);
    },
    function () {
        console.log('Completed');
    });

// => Next: 200
// => Next: 100
```

Common RxJS Operators

Support for common higher order functions from JavaScript

- map
- filter
- reduce

Common RxJS Operators

- merge
- take
- forkJoin

Additional Resources

- André Staltz RxJS Introduction https://gist.github.com/staltz/868e7e9bc2a7b8c1f754
- Ben Lesh RxJS course https://egghead.io/instructors/ben-lesh

Angular HTTP Service

> *Http is available as an injectable class, with methods to perform http requests*

Angular Cli

The Angular CLI is a command line interface tool that can create a project, add files, and perform a variety of ongoing development tasks such as testing, bundling, and deployment.

Angular Cli Installation

```
npm i -g @angular/cli
# yarn global add @angular/cli
```

Create a project

```
ng new --link-cli --routing --style scss my-project
```

```
├── e2e
├── package.json
├── protractor.conf.js
├── src
│   ├── app
│   │   ├── app-routing.module.ts
│   │   ├── app.component.ts
│   │   └── app.module.ts
│   ├── index.html
│   ├── main.ts
│   ├── polyfills.ts
│   ├── styles.scss
├── tsconfig.json
└── yarn.lock
```

Angular Cli common commands

- new
- serve (server)
- build
- generate
- test
- e2e
- lint

Demo

- Show parts of an Angular application
- Show examples of component types
- Show custom directives

Resources

- https://angular.io/docs/ts/latest/tutorial/
- https://angular.io/docs/ts/latest/guide/architecture.ht

Questions?

Thanks!

Slides available at https://cmckni3.github.io/angular-deep-dive-presentation Demo code available at https://github.com/cmckni3/angular-deep-dive-demo