Manchester Angular Workshop
Upgrading From AngularJS

darrylbrown.co.uk

= Consultdarryl

github.com/consultdarryl/manchester-angular-workshop

= ConsultDarryl

Why?

- Better performance (next slide)
- Less bugs
- Faster development (better tooling, refactoring, intellisense)
- Leverage cool new features
  - Universal, PWA, Native, Animation, Accessibility
- Future proof your website
- Less risky than a rewrite
- Upgrade your skills

git = ConsultDarryl

Performance

- Before (AngularJS 1.6)

28 requests | 1.4 MB transferred | Finish: 2.51 s | DOMContentLoaded: 2.09 s | Load: 2.53 s

- After (Angular 5.2)

21 requests | 1.9 MB transferred | Finish: 1.38 s | DOMContentLoaded: 1.18 s | Load: 1.41 s

- Approx. 35,000 lines of code
- 0.5MB = js-joda and momentjs (source-map-explorer)
  - Webpack can't tree shake these libraries

Approach

- ngUpgrade
  - Run Angular 1 and Angular 5 side-by-side
- Just do it!
  - Migrate to Angular 5

Preparation

- Upgrading is easier if…
  - You're using v1.6
  - Already following the AngularJS style guide
    - Convert directives/controllers into components

- Keep a copy of the original code in a temporary folder
  - (you will keep referring back to it)

= ConsultDarryl

How to

- Run **ng new** and build
- Rewrite index.html
  - Move .js and .css references into the **angular-cli.json**
  - Keep building to make sure everything is working
- Pick something simple and isolated e.g. a low level service
  - Copy the file to the new project
  - Rename .js to .ts
  - Change the decorators (next slide)
  - Register it in app.module.ts
  - Compile and fix
- Add routing
- Repeat

git = ConsultDarryl

Decorators

- Before (AngularJS 1.6)

```
angular
      .module('app.components.address', ['app.services.features', 'app.directives’])
      .component('addressComponent’, {
            templateUrl: './app-components/address/address.html’,
            bindings: {
                  address: '<‘,
                  requireEmail: '<‘,
                  onChange: ‘&’
            },
            controller: AddressController})
```

- After (Angular 5.2)

```
import { Component, OnInit } from '@angular/core’;
import { NgForm, NgModel } from '@angular/forms’;


@Component({
      selector: 'mb-address’,
      styleUrls: ['./address.component.less’],
      templateUrl: './address.component.html’,
      encapsulation: ViewEncapsulation.None})
export class AddressComponent implements OnInit {
```

git = ConsultDarryl

Afterwards

- Add types
- Break up components into modules
- Lazy load
- Fix file names e.g. error.component.ts
- Put everything into the correct folder structure
- Fix unit tests

git = ConsultDarryl

Lessons Learnt

- 3 weeks to migrate 35,000 lines of code.
- Nothing magical happens when you rename .js files to .ts. Nothing breaks.
- Once you start adding types you reveal tonnes of latent JS errors.
- Do one thing at a time.
- Leave tricky things until later e.g. Upgrading 3rd party libraries.
  - Comment out things which are tricky to fix.
  - Add "TODO" comments so you can go back and fix later.
- Don't worry about unit tests.
- Get a basic app up and running asap and then work through the bugs.
- Ditch the original routing and start again.
- Don't worry about lazy loading until everything else is working. Turn off minify/uglify etc.
- ViewEncapsulation.None solves a lot of styling issues.

Darryl Brown

Modernising Angular Websites

- Get in touch
  - darrylbrown.co.uk/contact
  - git = ConsultDarryl

git = ConsultDarryl