

GETTING STARTED

Ori Calvo, 2017

oric@trainologic.com

<http://trainologic.com>



Objectives

2

- Angular History
- Getting Started with Angular
- Identify Angular dependencies
- Develop basic Angular component
- Use @angular/cli

Industry Trends

3



Angular

4

- ❑ Almost 3 years of development
- ❑ Now at version 5
- ❑ AngularJS is based on concepts rooted at 2009
- ❑ Angular aims to “upgrade” AngularJS with new 2016/2017 concepts
- ❑ Not backward compatible
- ❑ Does support side by side execution with AngularJS

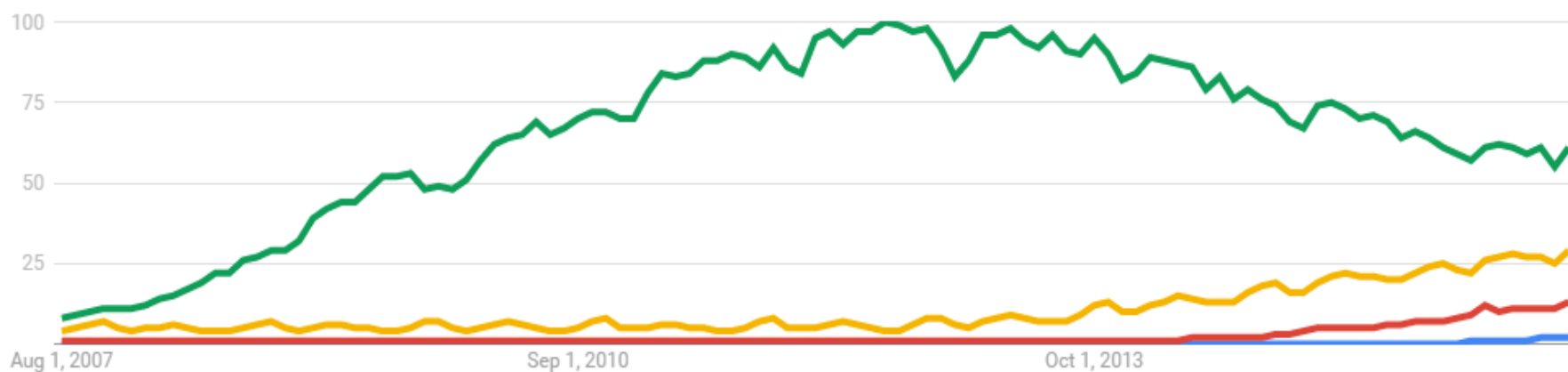
New Concepts

5

- ❑ Component based architecture
- ❑ Unidirectional data flow
- ❑ Server side rendering
- ❑ Running inside web workers
- ❑ Native development
- ❑ Pre compilation of views
- ❑ Observables
- ❑ Hierarchical Dependency Injection

Angular vs. Others

6



- ☐ jQuery
- ☐ Angular
- ☐ React
- ☐ Angular2

Getting Started

7

- The easiest way is to use **@angular/cli**
- Hold your horses ... lets do it manually
 - ▣ Module
 - ▣ Component
 - ▣ Bootstrapping
 - ▣ Polyfills
 - ▣ Typescript
 - ▣ Webpack

Installing Angular Dependencies

8

- Start with
 - ▣ `npm install @angular/platform-browser-dynamic`
- Fix all “UNMET PEER DEPENDENCY”
 - ▣ `@angular/platform-browser-dynamic`
 - ▣ `@angular/core`
 - ▣ `@angular/compiler`
 - ▣ `@angular/platform-browser`
 - ▣ `@angular/common`
 - ▣ `rxjs`
 - ▣ `zone.js`

Angular Polyfills

9

- Depends on your browser
- At minimum
 - ▣ reflect-metadata
 - Reflect API
 - ▣ zone.js
 - Not really a polyfill
 - Helps Angular handle asynchronous code

Angular “Minimal” Ingredients

10

- Module
- Component
- Bootstrapping

Angular Module

11

```
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { AppComponent } from './app.component';
4 import { ClockComponent } from './clock.component';
5
6 @NgModule({
7   imports: [ BrowserModule ],
8   declarations: [ AppComponent, ClockComponent ],
9   bootstrap: [ AppComponent ]
10 })
11 export class AppModule { }
```

Enjoy the public
content of other
modules

Make these
components
available to the
application

The component to
be loaded when
this module is
bootstrapped

Angular Module

12

- ❑ Consolidates components, directives and pipes into cohesive blocks of functionality
- ❑ Provides services
- ❑ Can be lazy loaded
- ❑ Usually per feature or per library
- ❑ Has public/private interfaces

Angular Component

13

Component metadata is injected using decorators

```
1 import {Component} from "@angular/core";
2
3 @Component({
4   selector: "my-app",
5   template: "<h1>Hello Angular 2</h1>"
6 })
7 export class AppComponent {
8 }
```

HTML element name

The template that will be injected into the component host element

Angular Component

14

- The term “controller” is no longer being used by Angular
 - ▣ Resembles the industry shift from MVC to component based architecture
- A component consist of
 - ▣ Name
 - ▣ Logic
 - ▣ Template
 - ▣ Styles
 - ▣ Metadata

Bootstrapping

15

Browser is not
the only
supported
platform

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';  
import { AppModule } from './app.module';  
  
platformBrowserDynamic().bootstrapModule(AppModule);
```

Why not just
name it
“bootstrap“ ?

Bootstrapping

16

- ❑ No automatic bootstrapping 😊
- ❑ You must tell Angular when to initialize the application
 - ▣ Allows for easier integration with 3rd party libraries
- ❑ Just like AngularJS you specify the root module and Angular does the magic

Configure Typescript

17

- Run `npm install typescript`
- Add `tsconfig.json`
- Angular depends on Typescript decorator support
 - ▣ `experimentalDecorators`
 - ▣ `emitDecoratorMetadata`
- Use `lib: ["dom", "es2015"]` to support standard libraries that Angular uses

tsconfig.json

18

Convert import
to require

Resolving non
relative modules
according to
NodeJS convention

Angular relies
heavily on
decorators
metadata

```
1 {  
2   "compilerOptions": {  
3     "target": "es5",  
4     "module": "commonjs",  
5     "moduleResolution": "node",  
6     "sourceMap": true,  
7     "emitDecoratorMetadata": true,  
8     "experimentalDecorators": true,  
9     "removeComments": false,  
10    "noImplicitAny": false  
11  }  
12 }
```

Compile Your code

19

- Typescript compiler is located under node_modules
 - ▣ `node_modules/.bin/tsc`
- Just execute it and it will read all options from the `tsconfig.json`
- Ensure you don't get any compilation errors
- You may add a package.json scripts command

```
1 {  
2   "scripts": {  
3     "tsc": "tsc"  
4   }  
5 }
```

Compiled main.js

20

```
1 "use strict";

2 var platform_browser_dynamic_1
  = require('@angular/platform-browser-dynamic');

3 var app_module_1
  = require('./app.module');

4 platform_browser_dynamic_1.platformBrowserDynamic()
  .bootstrapModule(app_module_1.AppModule);

5 /// sourceMappingURL=main.js.map
```

The Typescript
compiler uses
require instead
of import

Module Loader

21

- The keyword `import` is not yet supported by browsers
- We need to convert it to different syntax
- The common practice is to use **CommonJS** modules
- The Typescript compiler can transform “`import`” to “`require`”

Module Loader

22

- Two popular libraries for loading CommonJS modules inside the browser
 - ▣ **WebPack** – 8M downloads per month
 - ▣ **SystemJS** – 0.5M downloads per month
- Angular prefers Webpack
 - ▣ Many options
 - ▣ Large eco system
 - ▣ Too complex ☹

Webpack

23

- **npm install webpack**
- Create webpack.config.js

```
const path = require('path');

module.exports = {
  entry: './main.js',
  output: {
    filename: 'bundle.js',
    path: path.resolve(__dirname, 'dist')
  }
};
```

- **node_modules/.bin/webpack --watch**

index.html

24

- Webpack creates bundles under **dist** folder
- We need to manually include the bundles inside the HTML

```
<body>
  <app-root></app-root>

  <script src="dist/bundle.js"></script>
</body>
```

- Can be automated using **html-webpack-plugin**
 - ▣ Out of scope

@angular/cli

25

- ❑ Even when using **Webpack**, implementing build scripts is considered a complex task
- ❑ So the Angular team created an abstraction layer on top of Webpack
 - ❑ So now you need to learn both ...
- ❑ Starting with @angular/cli is easy
- ❑ At the long term you understand that customization capabilities resides inside Webpack and not inside angular/cli

@angular/cli

26

- ❑ Very opinionated
- ❑ A complete technology stack
- ❑ Strict directory structure
- ❑ Supports unit testing + E2E
- ❑ Development server
- ❑ Production build
- ❑ Scaffolding

@angular/cli Getting Started

27

- Install CLI tool globally
 - `npm install -g @angular/cli`
 - `yarn global add @angular/cli`
- Verify installation: `ng -v`
- Create new project

Create new project

28

- **ng new my-project**
- A new directory is created with all source files
 - package.json
 - tsconfig.json
 - .angular-cli.json
 - e2e – End to end testing
 - src/app – Component & Services
 - src/assets – Runtime assets
 - More ...

ng new options

29

- **--directory**: Name of directory to create, by default this is the application name
- **--prefix**: Component selector prefix
 - ▣ Can be overridden per component
- **--inline-style**: Do not generate CSS file
 - ▣ Can be overridden per component
- **--inline-template**: Do not use inline templates
 - ▣ Can be overridden per component

.angular-cli.json

30

- This is @angular/cli configuration file
- Use is to customize aspects of @angular/cli
- For example,
 - ▣ defaults/serve/port
 - ▣ apps[0]/prefix
 - ▣ app[0]/environments

ng serve

31


- ❑ Same as **npm start**
- ❑ Starts a development server on port 4200
- ❑ JavaScript bundles are created in memory
- ❑ Bundles are injected into **Index.html**
- ❑ Any change to the file system triggers re-build
- ❑ Use **--open** option to open a browser
 - ❑ Can fix the “npm start” command

--routing

32

- Commonly used cli command option to create a new project and automatically add a routing file in order to implement routing in angular app
- **ng new myapp --routing**

The project files tree after the command.
A routing module file is now available



```
src
-app
----app.component.css
----app.component.html
----app.component.spec.ts
----app.component.ts
----app.module.ts
----app-routing.module.ts
-assets
-environments
-favicon.ico
-index.html
-polyfills.ts
-main.ts
-styles.css
-test.ts
-tsconfig.app.json
-typings.d.ts
-tsconfig.spec.json
```


ng generate

33

- Assists in creating features to the app such as components, modules, services, pipes & directives
- Some options are derived from project level definition
- Some options can be re-defined
- Also have other options such as:
 - ▣ **--inline-template** use an inline template instead of a separate HTML file
 - ▣ **--inline-style** use inline styles instead of a separate CSS file
 - ▣ **--prefix** change prefix selector

--flat

34

- Do not generate a parent directory when generating a new component
- **ng g component contactList --flat**
- Probably you will want to use it when defining a new root component per feature module
 - ▣ To be consistent with app.component.ts

assets

35

- By default all static files are rejected
 - ▣ Except Webpack bundles
- Solution,
 - ▣ Put the asset inside the **assets** directory
 - ▣ The directory is part of production build
- In case of images consider using background-image
 - ▣ Thus the image is bundled

SCSS

36

- By default @angular/cli uses simple CSS files
- You may fix that
 - ▣ defaults/styleExt → **scss**
- You should also rename **app/styles.css**

src/style.css

37

- A global CSS that is injected into index.html
- Use it to
 - ▣ Define styling prior Angular load
 - ▣ Global application theme

More Commands

38

- <https://github.com/angular/angular-cli/wiki>
- ng lint
- ng test
- ng e2e
- ng build
- ng get/set
- ng eject

@angular/cli stories

39

- ❑ <https://github.com/angular/angular-cli/wiki/stories>
- ❑ HMR
- ❑ Proxy
- ❑ Routing
- ❑ Bootstrap
- ❑ Many more

Summary

40

- @angular/cli is an abstraction layer on top of Webpack
- As such it makes life easier (short term)
- Consider use **ng eject** and work directly with Webpack configuration