

Angular Tooling

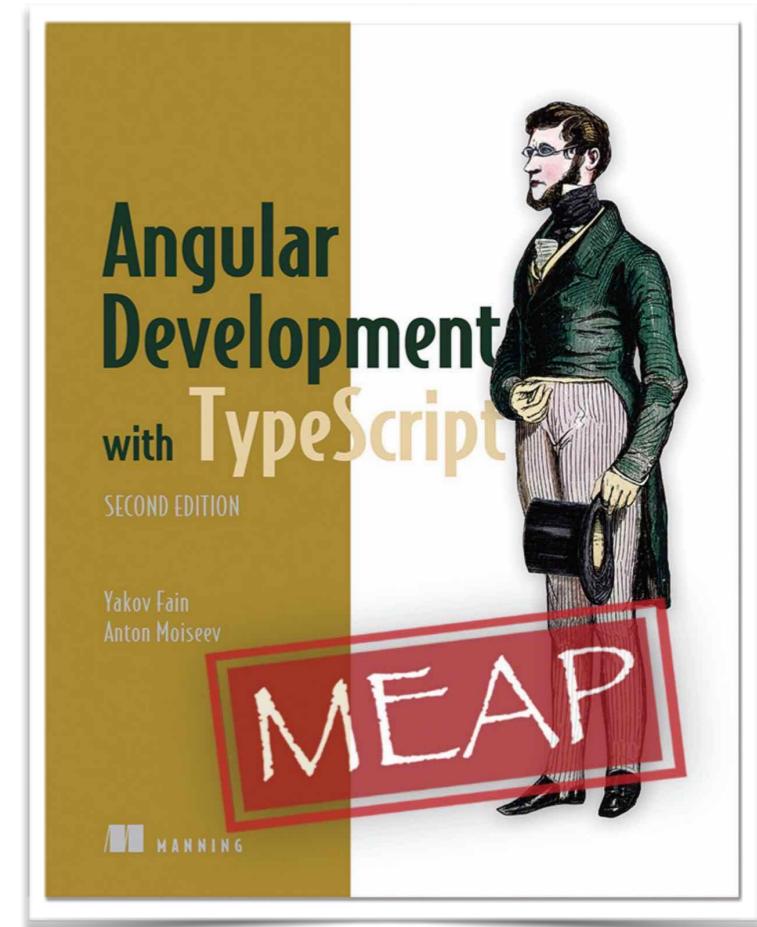
Yakov Fain, Farata Systems



@yfain

About myself

- Work for Farata Systems
- Angular consulting and training
- Java Champion
- Co-authored two editions of the book
“Angular Development with TypeScript”
- Working on a book
“Get Programming with TypeScript”



Code **ctwangsummit18**
40% off all books
at manning.com

@yfain

The Agenda

- Package managers
- Latest Angular CLI
- Configuring proxies
- Working with environment variables
- Build automation with npm scripts
- Using source maps

npm: Node Package Manager

- Install the latest version of Node.js from
<https://nodejs.org>
- npm comes with Node.js
- The npmjs.org repository has 400K+ packages

Semantic Versioning

6.0.1

Major
breaking
changes

Minor
new features,
not breaking

Patch
bug fixes,
not breaking

Installing packages with npm

- Project dependencies are configured in the file `package.json` and are installed locally in the directory `node_modules`
- Installing the package xyz in your project's dir `node_modules`:
`npm install xyz`
- Installing the package xyz globally:
`npm install xyz -g`

package.json

Required for dev and prod

```
"dependencies": {  
    "@angular/animations": "^6.0.0",  
    "@angular/common": "^6.0.0",  
    "@angular/compiler": "^6.0.0",  
    "@angular/core": "^6.0.0",  
    "@angular/forms": "^6.0.0",  
    "@angular/http": "^6.0.0",  
    "@angular/platform-browser": "^6.0.0",  
    "@angular/platform-browser-dynamic": "^6.0.0",  
    "@angular/router": "^6.0.0",  
    "core-js": "^2.5.4",  
    "rxjs": "^6.0.0",  
    "zone.js": "^0.8.26"  
},  
"devDependencies": {  
    "@angular/compiler-cli": "^6.0.0",  
    "@angular-devkit/build-angular": "~0.6.0",  
    "typescript": "~2.7.2",  

```

Required for dev

Yarn: an alternative to npm

- Yarn also uses package.json but works faster than npm
- **npm install yarn -g**
- To install project dependencies from package.json:
yarn install
- Set Yarn as default package manager for Angular CLI:
ng set -g packageManager=yarn

More Yarn commands

- Install the xyz package globally
`yarn add global xyz`
- Install xyz locally saving it in dependencies in package.json
`yarn add xyz`
- Install xyz locally saving it in devDependencies
`yarn add xyz -D`

yarn.lock

```
"@angular/forms@^4.0.0":  
  version "4.1.0"  
  resolved "https://registry.yarnpkg.com/@angular/forms/-/forms-4.1.0.tgz#8eae2a45c4ba064b377f9280e59c012b5dac6b80"  
  
"@angular/http@^4.0.0":  
  version "4.1.0"  
  resolved "https://registry.yarnpkg.com/@angular/http/-/http-4.1.0.tgz#7ba0c4d044dee964021b7cf19cb146a2c31577a5"  
  
"@angular/material@^2.0.0-beta.3":  
  version "2.0.0-beta.3"  
  resolved "https://registry.yarnpkg.com/@angular/material/-/material-2.0.0-beta.3.tgz#ec31dee61d7300ece28fee476852db236ded1e13"  
  
"@angular/platform-browser-dynamic@^4.0.0":  
  version "4.1.0"  
  resolved "https://registry.yarnpkg.com/@angular/platform-browser-dynamic/-/platform-browser-dynamic-4.1.0.tgz#0250d82d4abd36be60bb31fc7448ac6e28036690"  
  
"@angular/platform-browser@^4.0.0":  
  version "4.1.0"  
  resolved "https://registry.yarnpkg.com/@angular/platform-browser/-/platform-browser-4.1.0.tgz#b981386be1a36f2af7f0679447fd97b7267b25de"  
  
"@angular/router@^4.0.0":  
  version "4.1.0"  
  resolved "https://registry.yarnpkg.com/@angular/router/-/router-4.1.0.tgz#dd3563662f95ca3aa3dd9ff13c6ed4bea1d90b06"
```

To ensure that your team uses the same packages,
keep yarn.lock in your version control repo

ES6 modules vs Angular modules

- **ES6 module** is a file with the code that imports or exports something
- **Angular module** lists the components, other modules, service providers, directives, and pipes that belong together

IDEs

Visual Studio Code: <https://code.visualstudio.com>

WebStorm: <https://www.jetbrains.com/webstorm>

StackBlitz (online VS Code): <https://stackblitz.com>

Angular CLI

Angular CLI features

- Scaffolding the project and creating a basic app
- Generating components, services, modules, etc.
- Dev web server
- Supports dev and prod builds
- Configuring test runners

Starting a new project

1. Install Angular CLI using the command window:

```
npm i @angular/cli -g
```

2. Generate a new Angular project **hello-cli**:

```
ng new hello-cli
```

3. **cd hello-cli**

4. Build the app bundles and start the app

```
ng serve -o
```

Selecting new options

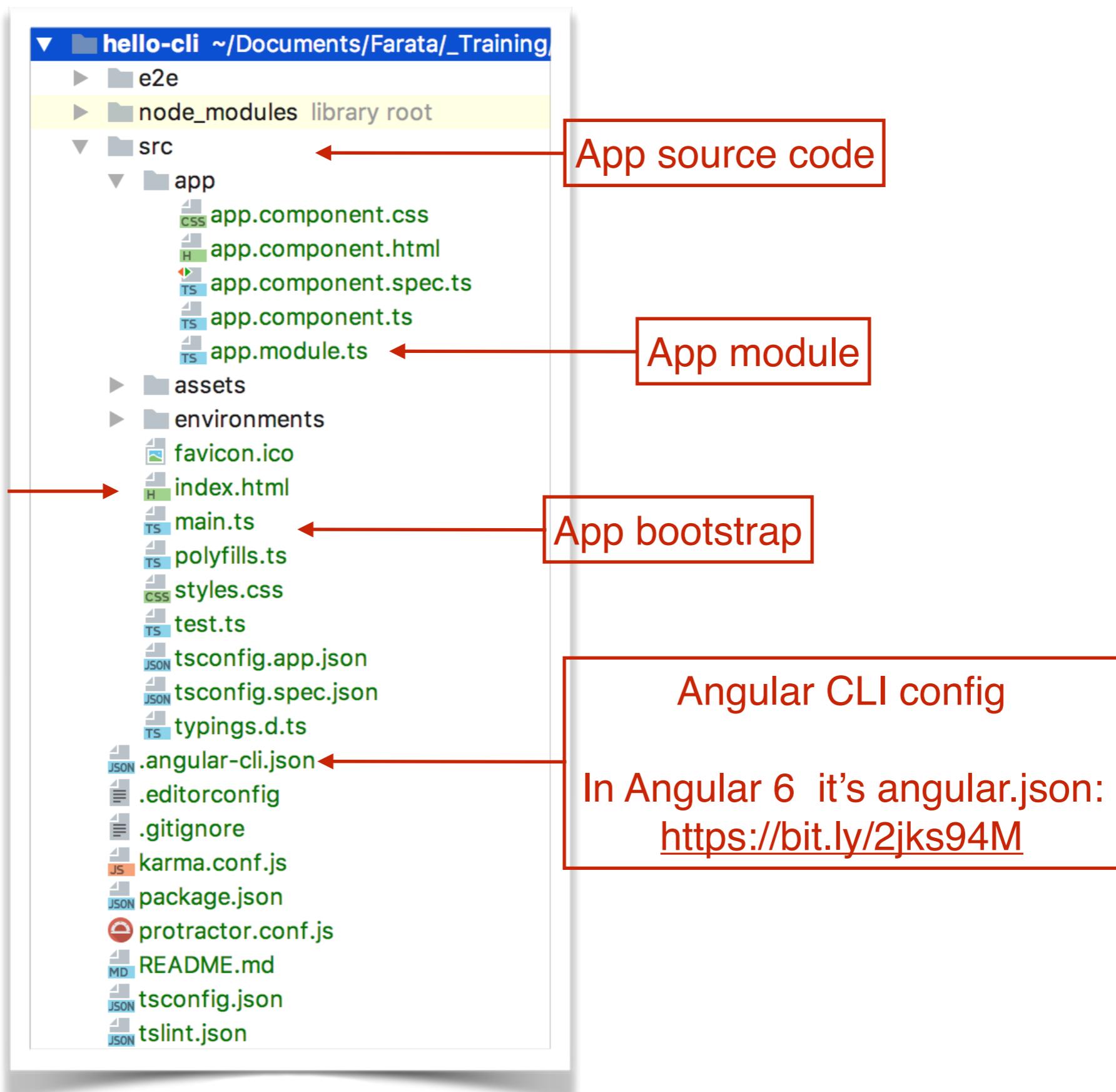
--routing

--inline-style

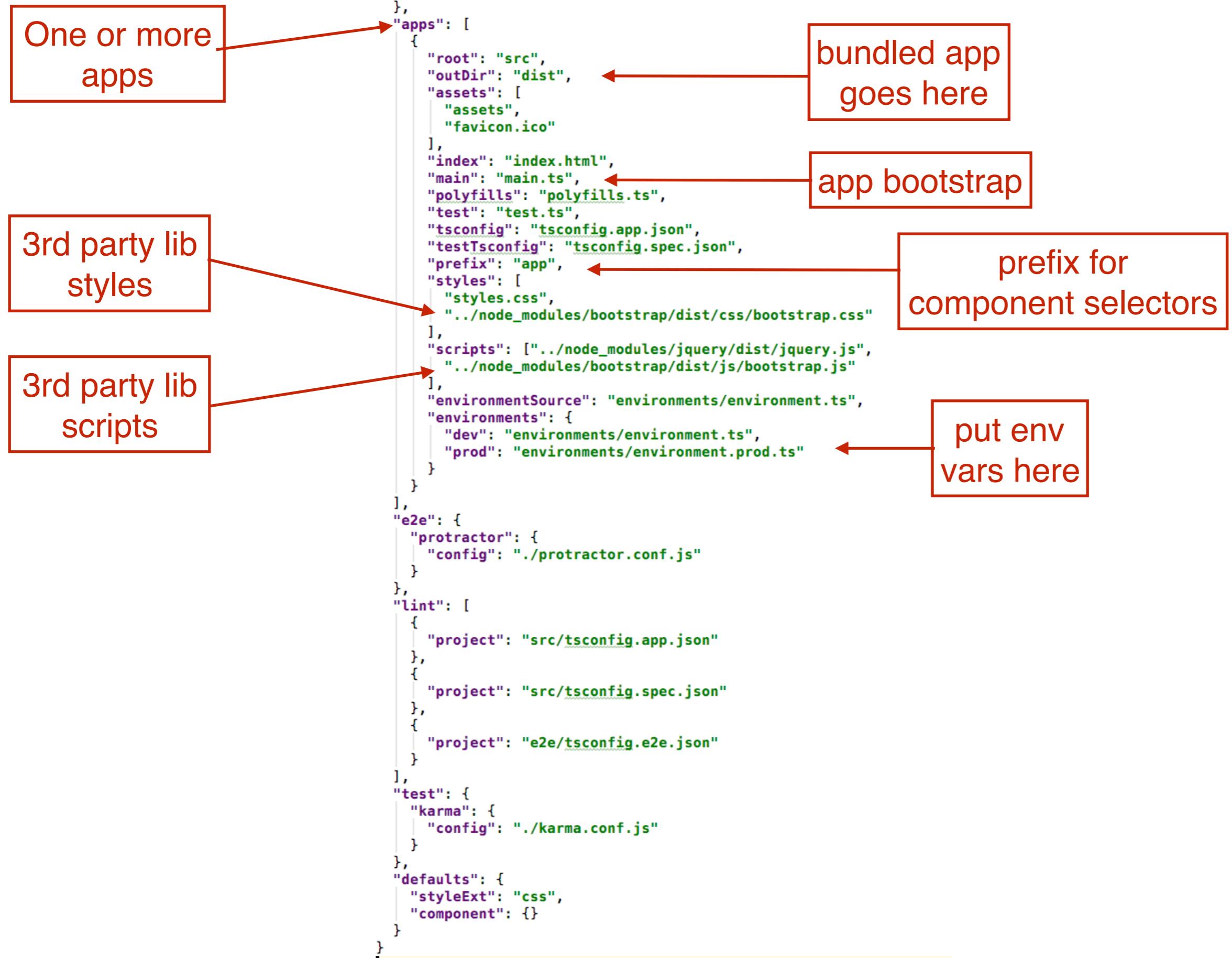
--inline-template

--skip-tests

The structure of the generated project



.angular-cli.json



The workspace file angular.json

One or more projects

```
{  
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",  
  "version": 1,  
  "newProjectRoot": "projects",  
  "projects": {  
    "hello-cli": {  
      "root": "",  
      "sourceRoot": "src",  
      "projectType": "application",  
      "prefix": "app",  
      "schematics": {},  
      "architect": {  
        "build": {  
          "builder": "@angular-devkit/build-angular:browser",  
          "options": {  
            "outputPath": "dist/hello-cli",  
            "index": "src/index.html",  
            "main": "src/main.ts",  
            "polyfills": "src/polyfills.ts",  
            "tsConfig": "src/tsconfig.app.json",  
            "assets": [  
              "src/favicon.ico",  
              "src/assets"  
            ],  
            "styles": [  
              "src/styles.css"  
            ],  
            "scripts": []  
          },  
          "configurations": {  
            "production": {  
              "fileReplacements": [  
                {  
                  "replace": "src/environments/environment.ts",  
                  "with": "src/environments/environment.prod.ts"  
                }  
              ],  
              "optimization": true,  
              "outputHashing": "all",  
              "sourceMap": false,  
              "extractCss": true,  
              "namedChunks": false,  
              "aot": true,  
              "extractLicenses": true,  
              "vendorChunk": false,  
              "buildOptimizer": true  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

app or library

bundled app goes here

Configuring environments

Creating a single project

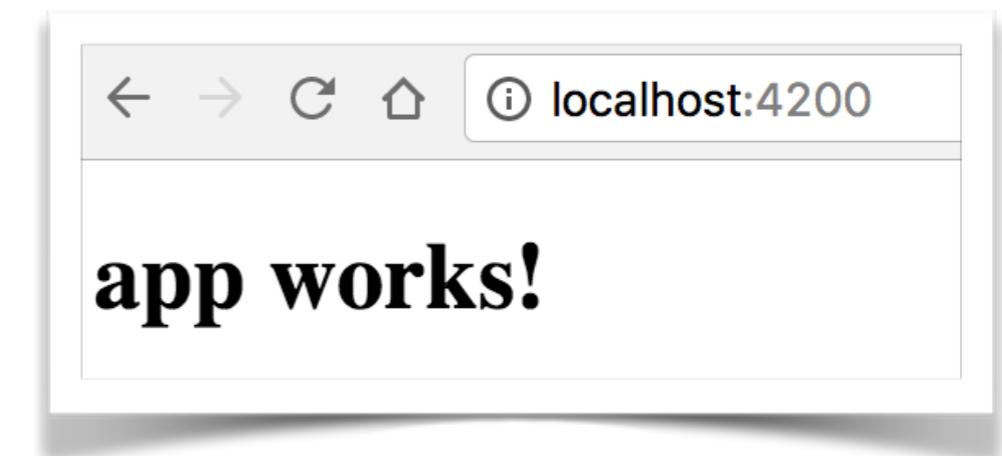
1. Generate a new Angular project `hello-cli` with the routing module

```
ng new hello-cli --routing
```

2. `cd hello-cli`

3. Build the app bundles in memory, and open the app in the browser:

```
ng serve -o
```



ng generate (ng g)

- ng g m shipping
 - ng g c product
 - ng g c product --module=shipping.module.ts
 - ng g c product -is -it -spec false
 - ng g s product
 - ng g application <appName>
- if an app has more than one module
- Inline styles,
inline templates,
no tests
- Generate an app
within the works[ace]

@Injectable and providedIn

ng g s product

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

@Injectable({
  providedIn: 'root'
})
export class ProductService {

  constructor() { }
}
```



No need to specify the provider for ProductService in the module

Creating a multi-app workspace

1. Generate a new Angular project myWorkspace

```
ng new myWorkspace
```

2. cd myWorkspace

3. Generate app1

```
ng g application app1
```

4. Generate app2

```
ng g application app2
```

5. Build and start the app you need, e.g.:

```
ng serve app1
```

6. Test the app you need, e.g.:

```
ng test app1
```

```
ng e2e app1
```

<https://update.angular.io>

Secure | https://update.angular.io

Angular Update Guide - 5.2 → 6.0 for Medium Apps

Select the options matching your project

Angular from to
5.2 6.0

How complex is
your app? Medium

ngUpgrade I use ngUpgrade

Package manager yarn

Show me how to update!

Angular Update Guide - 5.2 → 6.0 for Medium Apps

Before updating

- If you import any animations services or tools from `@angular/core`, you should import them from `@angular/animations`
- Switch from `HttpModule` and the `Http` service to `HttpClientModule` and the

Angular CLI: recent additions

- `ng update` - updates dependencies in package.json to the latest version of Angular and might update your code.

No guarantees that your project will compile.

Angular CLI: recent additions

- `ng update` - updates dependencies in package.json to the latest version of Angular and might update your code.
No guarantees that your project will compile.
- `ng new library <name>` - generates a project for creating a library instead of an app.

Angular CLI: recent additions

- `ng update` - updates dependencies in package.json to the latest version of Angular and might update your code.
No guarantees that your project will compile.
- `ng new library <name>` - generates a project for creating a library instead of an app.
- **`ng add`** - adds a package to your project and modifies some files in your project, e.g. `ng add @angular/material`

Setting budgets for prod builds

The screenshot shows a code editor and a terminal window.

Code Editor: The left pane shows a file tree of a Angular project directory. The right pane displays the contents of the `.angular-cli.json` file. A red arrow points to the `"budgets": [{}]` section.

```
6   "apps": [
7     {
8       "name": "formcontrol",
9       "root": "src",
10      "outDir": "dist",
11      "assets": [
12        "assets",
13        "favicon.ico"
14      ],
15      "budgets": [{}]
16        "type": "allScript",
17        "maximumWarning": "120kb",
18        "maximumError": "140kb"
19    },
20  ]
```

Terminal: The terminal shows the build output and resulting budgets.

```
+ Hash: f2f1cd4d562c973c1ebb
x Time: 18800ms
chunk {0} main.bd7438eab49df4e39fcd.bundle.js (main) 261 kB [initial] [rendered]
chunk {1} polyfills.12938b5564d86620a6f8.bundle.js (polyfills) 64.4 kB [initial] [rendered]
chunk {2} styles.ac89bfdd6de82636b768.bundle.css (styles) 0 bytes [initial] [rendered]
chunk {3} inline.318b50c57b4eba3d437b.bundle.js (inline) 796 bytes [entry] [rendered]

WARNING in budgets, maximum exceeded for total scripts. Budget 120 kB was exceeded by 206 kB.

ERROR in budgets, maximum exceeded for total scripts. Budget 140 kB was exceeded by 186 kB.
```

Three red arrows point from the terminal output to the `maximumWarning` and `maximumError` fields in the configuration file, highlighting the budget values being exceeded.

Docs: <https://bit.ly/2jhuq0D>

Polyfills for Internet Explorer

The image shows a code editor interface with a sidebar on the left displaying the project structure and a main editor area on the right showing the content of the `polyfills.ts` file.

Project Structure:

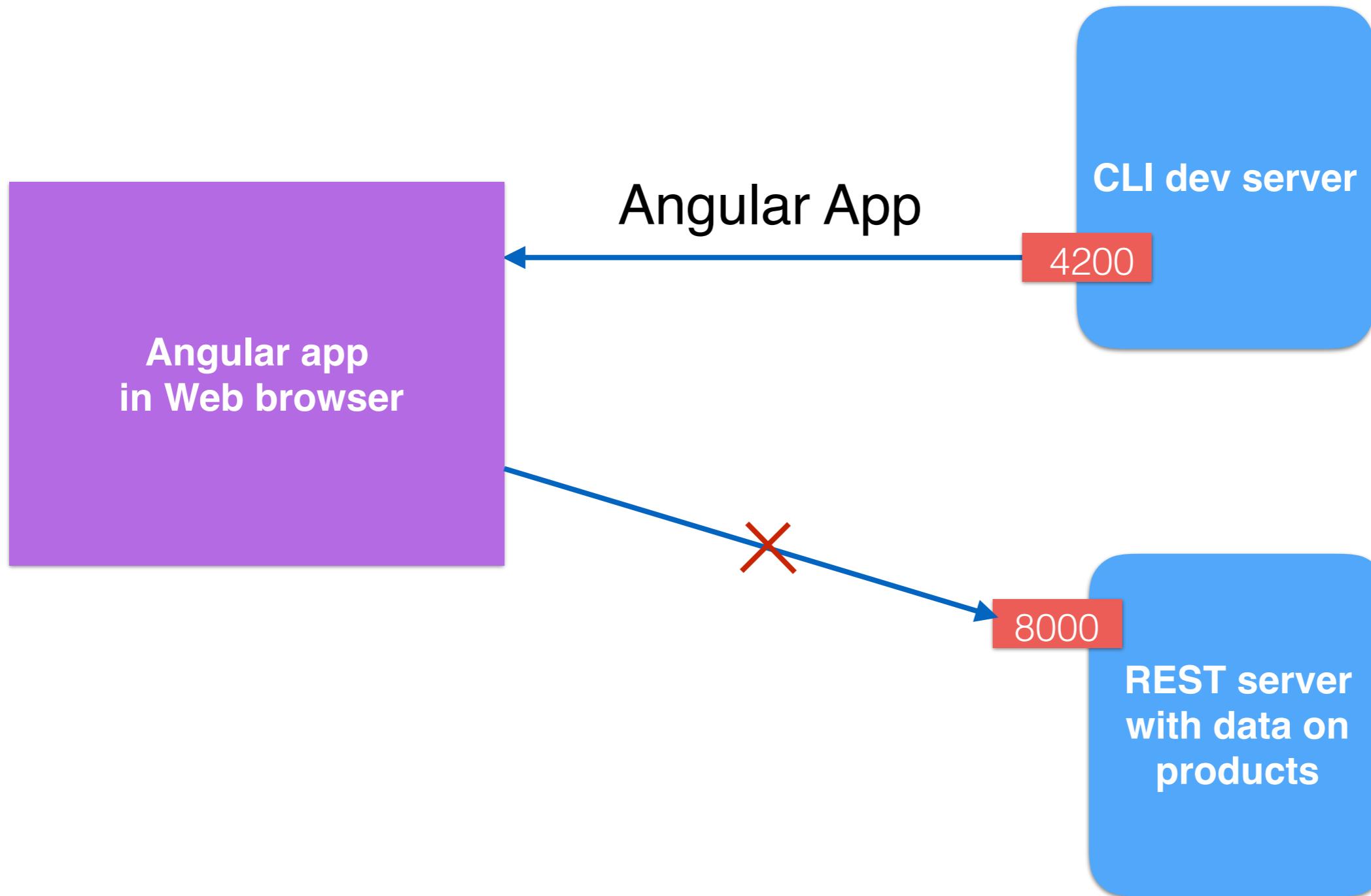
- src**
 - app
 - assets
 - data
 - environments
 - styles
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts**
 - test.ts
 - tsconfig.app.json
 - tsconfig.spec.json
 - typings.d.ts
- .angular-cli.json
- .editorconfig
- .gitignore
- karma.conf.js
- package.json
- protractor.conf.js
- tsconfig.json
- tslint.json
- yarn.lock

External Libraries

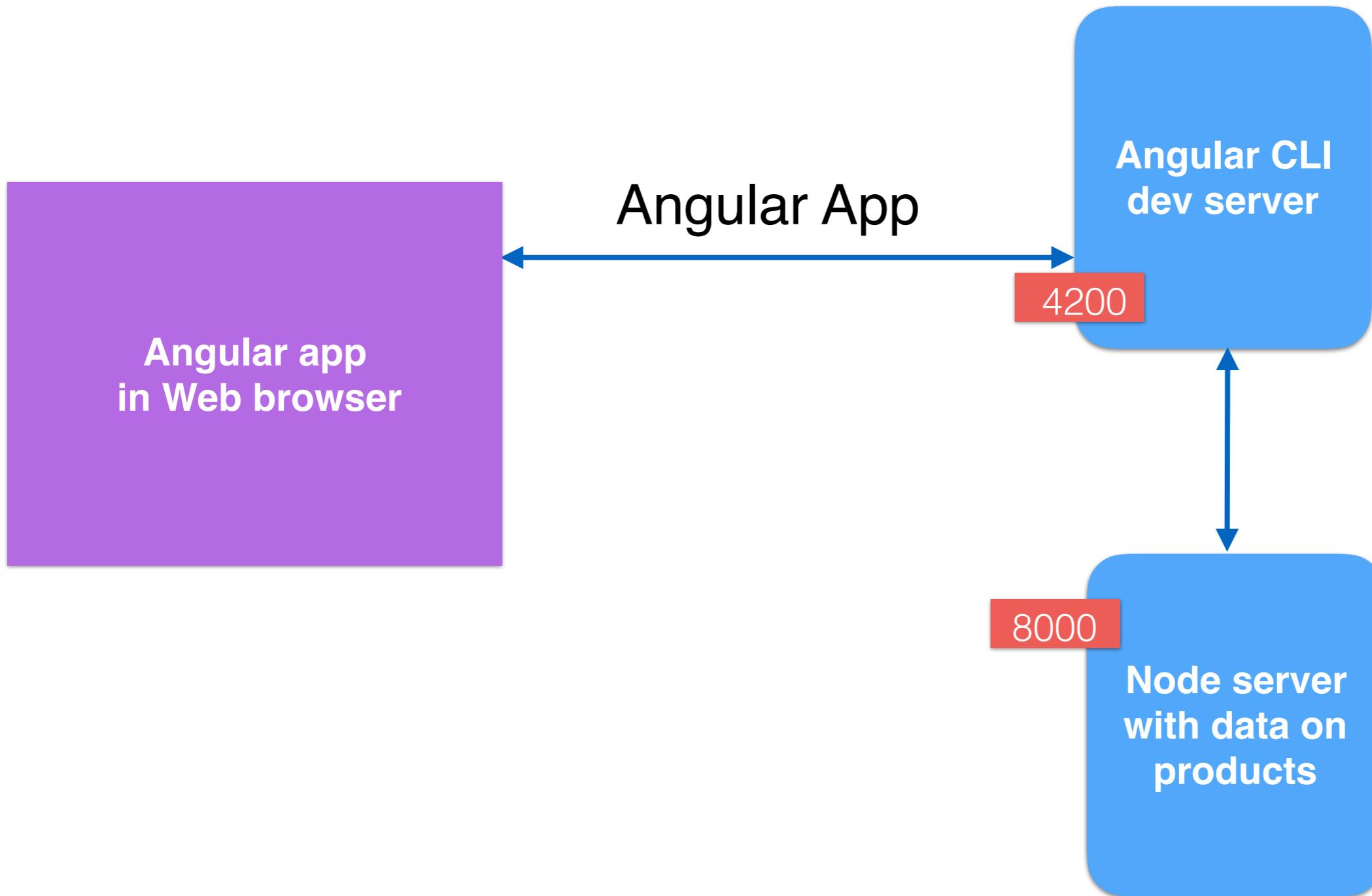
Content of polyfills.ts:

```
16
17 // *****
18 * BROWSER POLYFILLS
19 */
20
21 /** IE9, IE10 and IE11 requires all of the following imports. */
22 // import 'core-js/es6/symbol';
23 // import 'core-js/es6/object';
24 // import 'core-js/es6/function';
25 // import 'core-js/es6/parse-int';
26 // import 'core-js/es6/parse-float';
27 // import 'core-js/es6/number';
28 // import 'core-js/es6/math';
29 // import 'core-js/es6/string';
30 // import 'core-js/es6/date';
31 // import 'core-js/es6/array';
32 // import 'core-js/es6/regexp';
33 // import 'core-js/es6/map';
34 // import 'core-js/es6/weak-map';
35 // import 'core-js/es6/set';
36
37 /** IE10 and IE11 requires the following imports. */
38 // import 'classlist.js'; // Run `npm install --save classlist.js`
39
40 /** IE10 and IE11 requires the following imports. */
41 // import 'core-js/es6/reflect';
```

Two servers in dev mode



Two servers and a proxy (dev mode)



Same origin error

ng serve starts a dev server on port 4200

The REST server runs on port 8000

If the Angular app uses

`http.get ('http://localhost:8000/api/products')`,
it'll get this:

- ✖ XMLHttpRequest cannot load http://localhost:8000/api/products. No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://localhost:4200' is therefore not allowed access.

Configuring proxy for Angular client

```
proxy-conf.json
{
  "/api": {
    "target": "http://localhost:8080",
    "secure": false
  }
}
```

The Spring Boot
server runs here

```
Angular client: http.get('/api/products');
```

goes to 4200
and redirected
to 8080

```
ng serve --proxy-config proxy-conf.json
```

Building apps for prod
deployment

JiT vs AoT compilation

- **Just-in-Time** compilation: your app includes Angular's compiler that compiles the app in the browser.
- **Ahead-of-Time** compilation: Angular components and templates are precompiled into JS with the `ngc` compiler.
- The AoT-compiled apps don't include the Angular compiler

ng build for dev and prod

the output goes into the **dist** dir

- ng build



```
136B Jan 29 12:46 data
5.3K Jan 29 12:46 favicon.ico
609B Jan 29 12:46 index.html
5.7K Jan 29 12:46 inline.bundle.js
5.8K Jan 29 12:46 inline.bundle.js.map
9.0K Jan 29 12:46 main.bundle.js
6.2K Jan 29 12:46 main.bundle.js.map
212K Jan 29 12:46 polyfills.bundle.js
259K Jan 29 12:46 polyfills.bundle.js.map
11K Jan 29 12:46 styles.bundle.js
15K Jan 29 12:46 styles.bundle.js.map
2.4M Jan 29 12:46 vendor.bundle.js
2.9M Jan 29 12:46 vendor.bundle.js.map
```

- ng build --prod

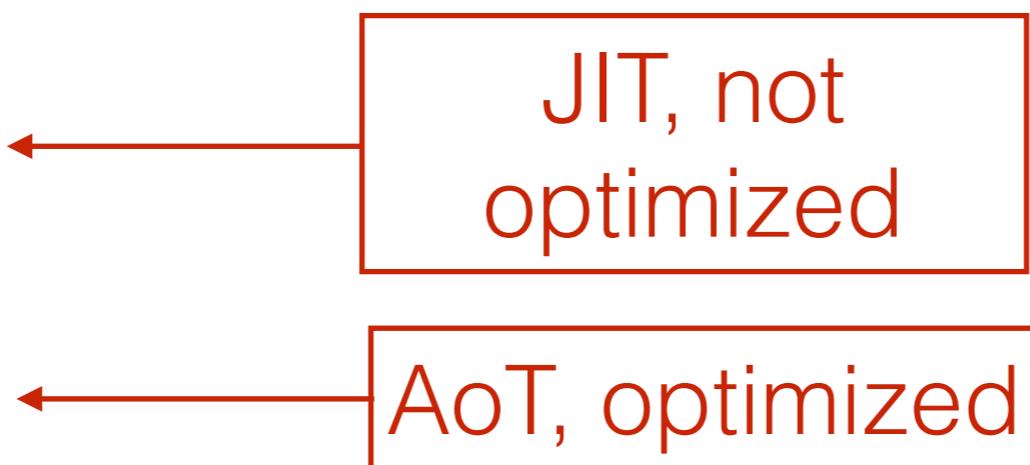
performs AoT
by default



```
3.2K Jan 29 12:51 3rdpartylicenses.txt
136B Jan 29 12:51 data
5.3K Jan 29 12:51 favicon.ico
589B Jan 29 12:51 index.html
1.4K Jan 29 12:51 inline.d483d84aa7d8440978f5.bundle.js
175K Jan 29 12:51 main.8522776bac4edaecdaad.bundle.js
64K Jan 29 12:51 polyfills.47853ebf6acf9efe05b4.bundle.js
79B Jan 29 12:51 styles.9c0ad738f18adc3d19ed.bundle.css
```

ng build

1. ng build
2. ng build --prod



Peeking inside the bundles

- **Source-Map-Explorer** analyzes the bundle's content using source maps
- `npm install -g source-map-explorer`
- `source-map-explorer main.bundle.js`

You can get a report in html, json, or tab-delimited formats.

```
ng build --prod --aot=false
```

/ • 1.27 MB • 100.0%

~ • 1.27 MB • 100.0%

@angular • 1.06 MB • 83.2%

compiler • 346 kB • 27.2%

@angular • 346 kB • 27.2%

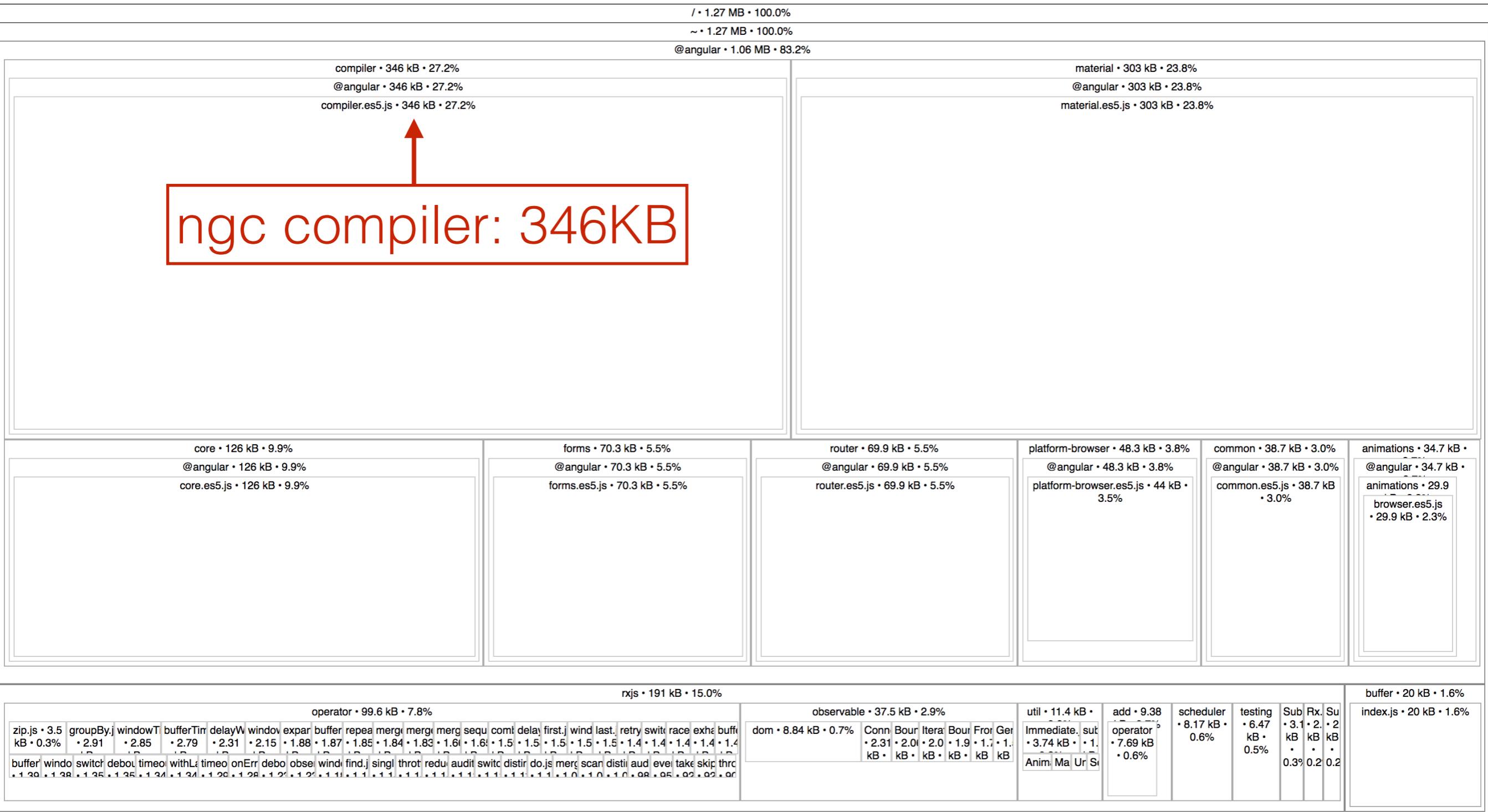
compiler.es5.js • 346 kB • 27.2%

material • 303 kB • 23.8%

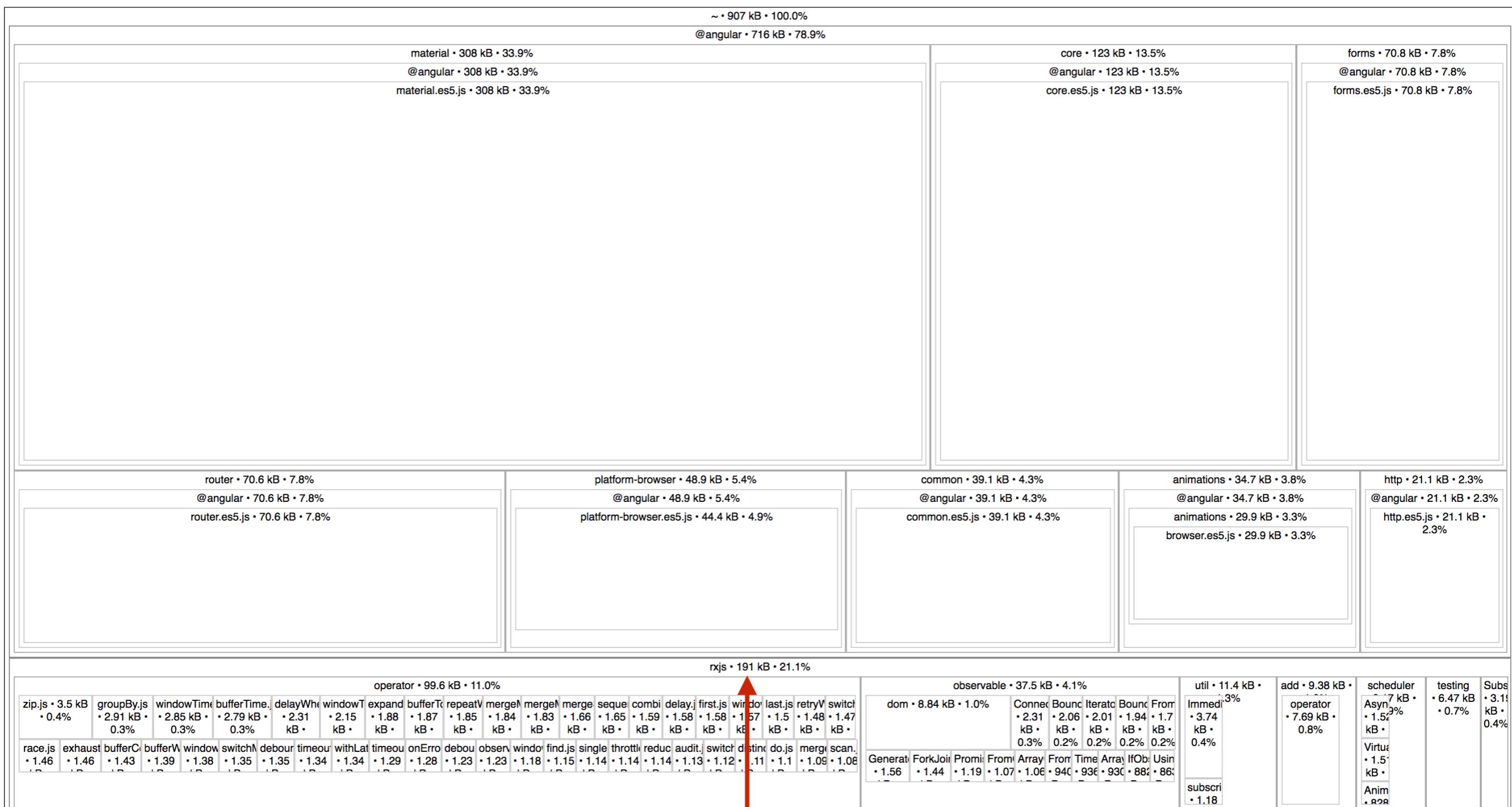
@angular • 303 kB • 23.8%

material.es5.js • 303 kB • 23.8%

ngc compiler: 346KB



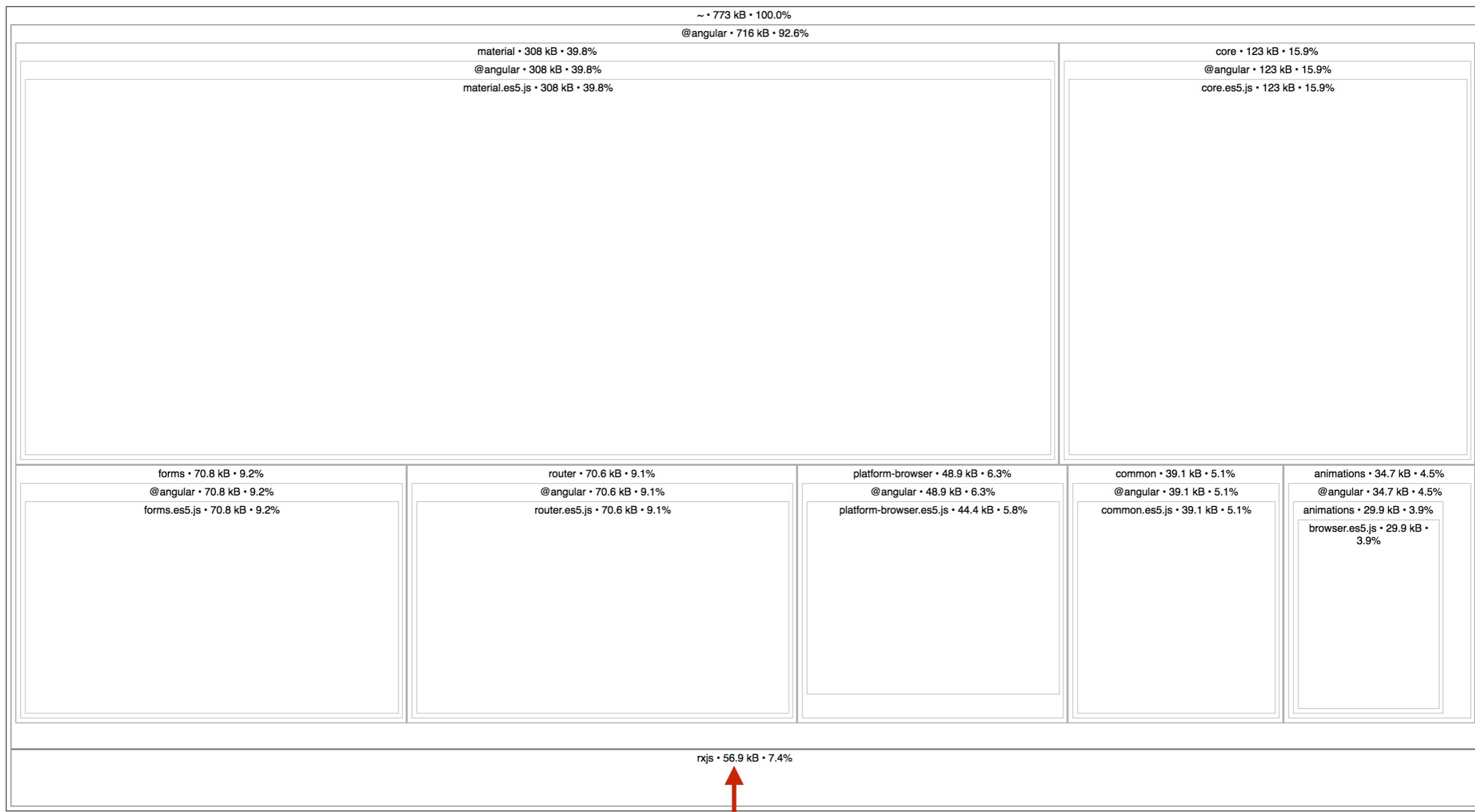
ng build --prod



RxJS: 191KB

In Angular 5, this was the issue: import { Observable } from 'rxjs'

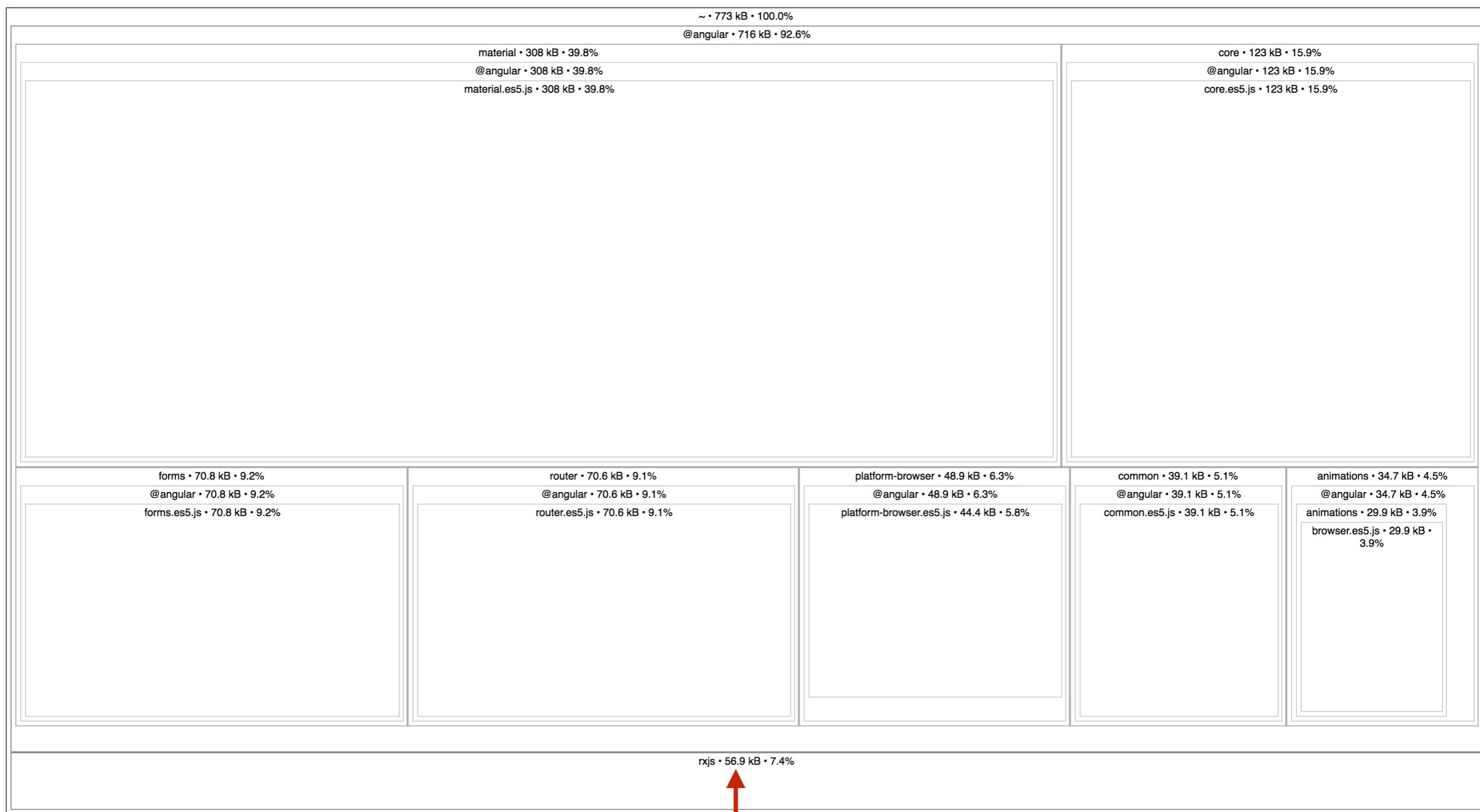
After fixing imports: ng build --prod



RxJS: 57KB with fixed import:

Angular 5: `import { Observable } from 'rxjs/Observable'`

ng build -prod

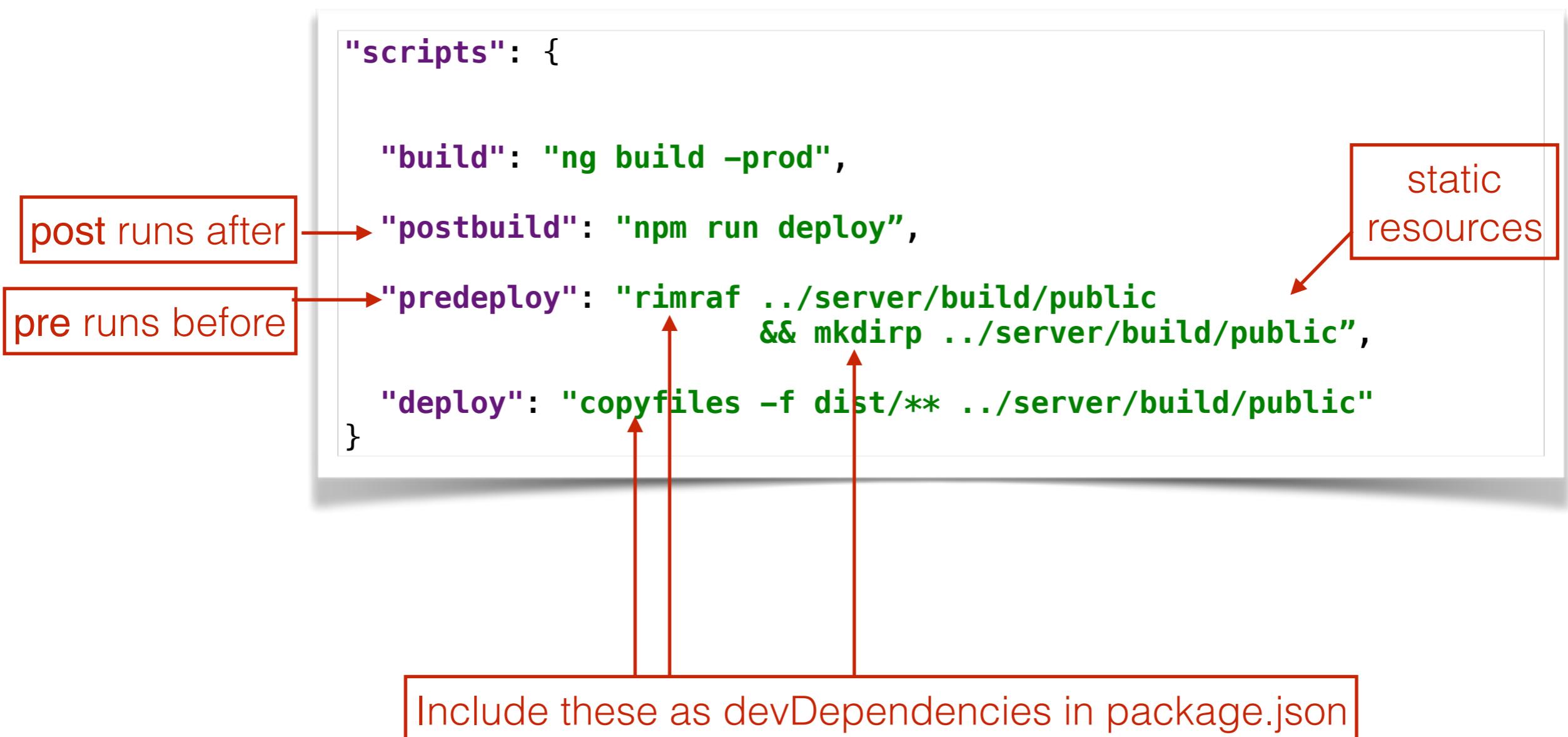


RxJS: 57KB with fixed import:

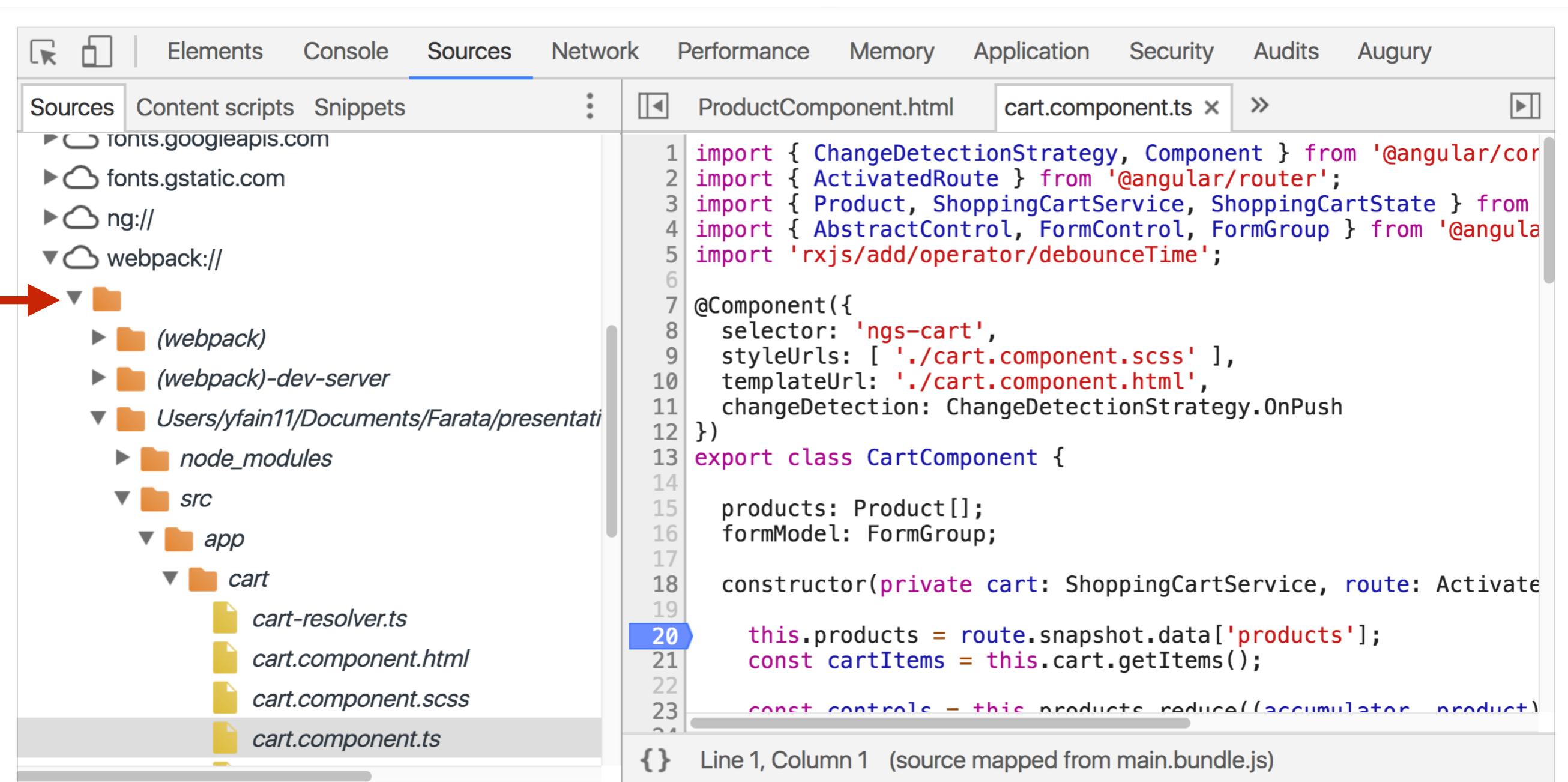
~~import { Observable } from 'rxjs/Observable'~~

Angular 6: import { Observable } from 'rxjs'

Automating deployment with npm scripts



Debugging TypeScript in the browser



The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. The left sidebar displays a tree view of source files, with a red arrow pointing to the 'src/app/cart' folder. The main panel shows the content of 'cart.component.ts'. The code is annotated with line numbers and color-coded imports. A blue box highlights line 20, which contains the assignment 'this.products = route.snapshot.data['products'];'. The status bar at the bottom indicates 'Line 1, Column 1 (source mapped from main.bundle.js)'.

```
1 import { ChangeDetectionStrategy, Component } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3 import { Product, ShoppingCartService, ShoppingCartState } from '@ngf-farata';
4 import { AbstractControl, FormControl, FormGroup } from '@angular/forms';
5 import 'rxjs/add/operator/debounceTime';

6
7 @Component({
8   selector: 'ngs-cart',
9   styleUrls: [ './cart.component.scss' ],
10  templateUrl: './cart.component.html',
11  changeDetection: ChangeDetectionStrategy.OnPush
12 })
13 export class CartComponent {
14
15   products: Product[];
16   formModel: FormGroup;
17
18   constructor(private cart: ShoppingCartService, route: ActivatedRoute) {
19
20     this.products = route.snapshot.data['products'];
21     const cartItems = this.cart.getItems();
22
23     const controls = this.products.reduce((accumulator, product) => {
24       const control = new FormGroup({
25         quantity: new FormControl(product.quantity),
26         price: new FormControl(product.price)
27       });
28       accumulator.push(control);
29     }, new FormGroup());
30
31     this.formModel = controls;
32   }
33
34 }
```

{ } Line 1, Column 1 (source mapped from main.bundle.js)

Using env vars defined in environment files

environment.ts

```
export const environment = {  
  production: false,  
  baseHref: '/',  
  apiBaseUrl: 'http://localhost:8080',  
  oauthUrl: 'http://localhost:9090'  
};
```

main.ts

```
import { environment }  
from './environments/environment';  
  
if (environment.production) {  
  enableProdMode();  
}
```

environment.prod.ts

```
export const environment = {  
  production: true,  
  baseHref: '/auction',  
  apiBaseUrl: 'https://api.online.auction',  
  oauthUrl: 'https://oauth.online.auction'  
};
```

Build targets and environments

ng serve -prod

target

ng build -dev

ng build -prod



environment.prod.ts is used for the target prod

environment.ts is for t dev.

Adding custom environments

```
"environmentSource": "environments/environment.ts",
"environments": {
  "dev": "environments/environment.ts",
  "prod": "environments/environment.prod.ts",
  "qa": "environments/environment.qa.ts"
}
```



How to build

ng serve --dev -e=qa

Which env to use

ng serve --prod -e=qa

Karma Test Runner

- In Angular CLI projects:
`ng test`
- Add Karma launchers for multiple browsers
- Run tests as a part of your automated build

```
"scripts": {  
  "build": "ng build --prod && ng test"  
}
```



karma.conf

```
module.exports = function (config) {
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular/cli'],
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
      require('karma-jasmine-html-reporter'),
      require('karma-coverage-istanbul-reporter'),
      require('@angular/cli/plugins/karma')
    ],
    client:{
      clearContext: false // leave Jasmine Spec Runner output visible in browser
    },
    files: [
      { pattern: './src/test.ts', watched: false }
    ],
    preprocessors: {
      './src/test.ts': ['@angular/cli']
    },
    mime: {
      'text/x-typescript': ['ts','tsx']
    },
    coverageIstanbulReporter: {
      reports: [ 'html', 'lcovonly' ],
      fixWebpackSourcePaths: true
    },
    angularCli: {
      config: './angular-cli.json',
      environment: 'dev'
    },
    reporters: config.angularCli && config.angularCli.codeCoverage
      ? ['progress', 'coverage-istanbul']
      : ['progress', 'kjhtml'],
    port: 9876,
    colors: true,
    LogLevel: config.LOG_INFO,
    autoWatch: true,
    browsers: ['Chrome'],
    singleRun: false
  });
};
```

Adding Firefox to Karma config

Install the Karma plugin for Firefox

```
npm i karma-firefox-launcher --save-dev
```

In karma.conf.js add this line to the plugins section:

```
require('karma-firefox-launcher')
```

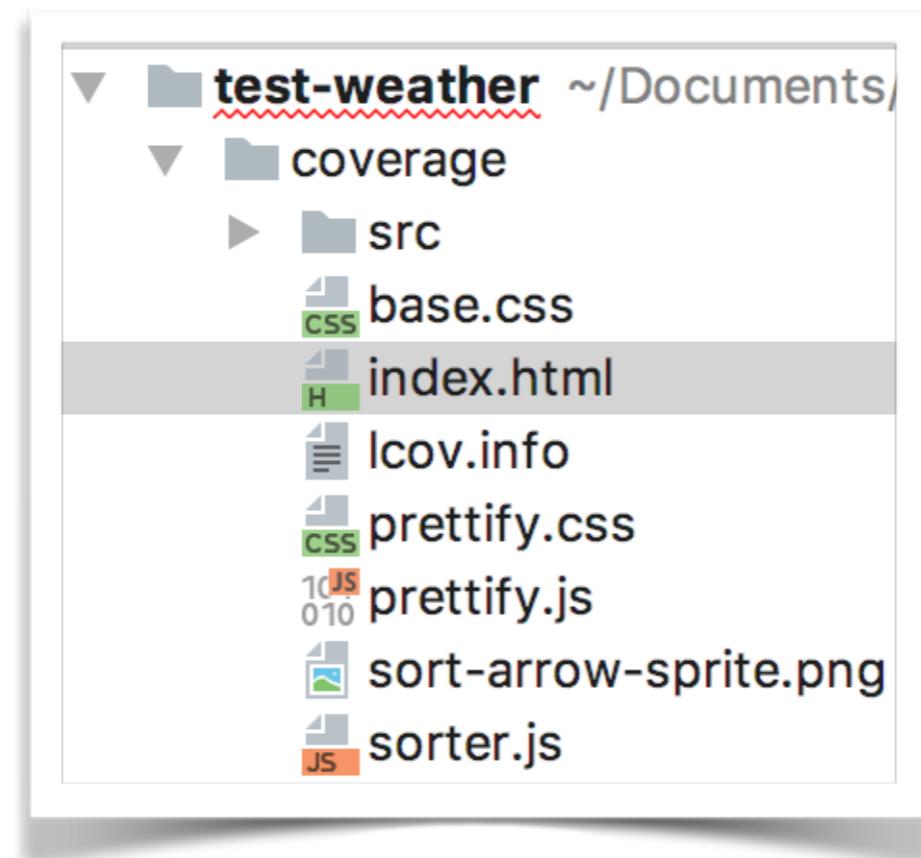
In karma.conf.js, add Firefox to the list of browsers:

```
browsers: ['Chrome', 'Firefox']
```

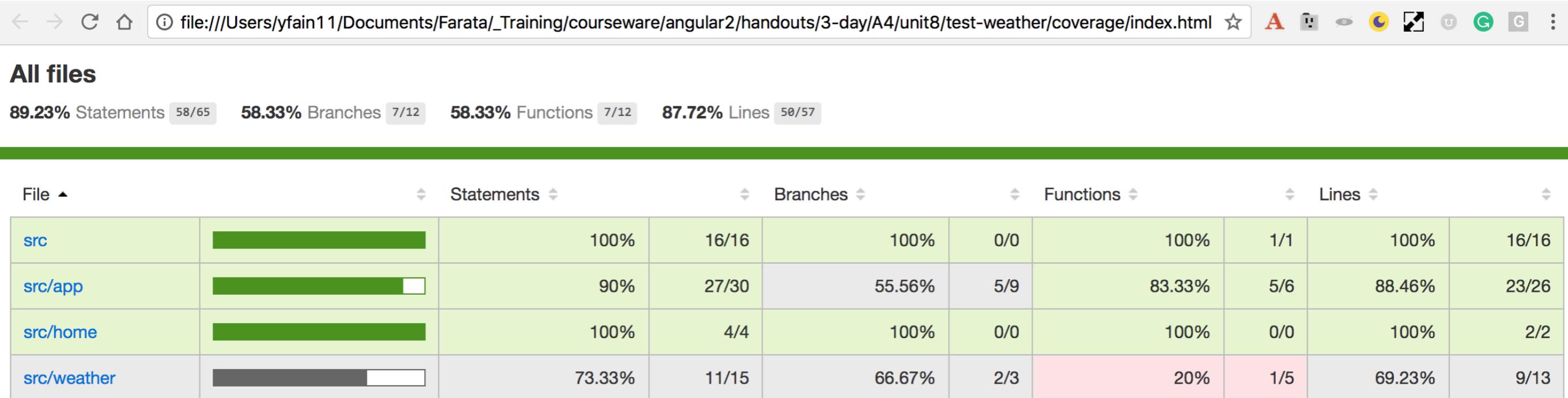
Code coverage

Generate code coverage report:
ng test --code-coverage

Open index.html in a newly created coverage folder



Code coverage report



Enforcing minimal code coverage

The screenshot shows a code editor with a file tree on the left and a code editor pane on the right. The file tree includes files like tsconfig.spec.json, typings.d.ts, .angular-cli.json, .editorconfig, .gitignore, karma.conf.js (which is selected), package.json, protractor.conf.js, README.md, tsconfig.json, tslint.json, and yarn.lock. The code editor pane displays the karma.conf.js configuration file with a yellow highlight on the 'functions: 80' line. The terminal window below shows the command 'ng test --watch=false --code-coverage' being run, followed by a series of error messages from the karma reporter indicating that coverage thresholds are not met for various metrics across multiple files.

```
tsconfig.spec.json
typings.d.ts
.angular-cli.json
.editorconfig
.gitignore
karma.conf.js
package.json
protractor.conf.js
README.md
tsconfig.json
tslint.json
yarn.lock
External Libraries

coverageIstanbulReporter: {
  reports: ['html', 'lcovonly'],
  fixWebpackSourcePaths: true,
  thresholds: {
    global: { // thresholds for all files
      statements: 80,
      lines: 80,
      branches: 80,
      functions: 80
    },
    each: { // thresholds per file
      statements: 80,
      lines: 80,
      branches: 80,
      functions: 80
    }
  }
}

MacBook-Pro-5:test-weather yfain11$ ng test --watch=false --code-coverage
05 05 2017 06:03:45.459:INFO [karma]: Karma v1.4.1 server started at http://0.0.0.0:9876/
05 05 2017 06:03:45.462:INFO [launcher]: Launching browser Chrome with unlimited concurrency
05 05 2017 06:03:45.465:INFO [launcher]: Starting browser Chrome
05 05 2017 06:03:46.637:INFO [Chrome 57.0.2987 (Mac OS X 10.11.6)]: Connected on socket NRiMBxbba36Pe7x5iAAAAA
Chrome 57.0.2987 (Mac OS X 10.11.6): Executed 5 of 5 SUCCESS (0.372 secs / 0.357 secs)
05 05 2017 06:03:47.727:ERROR [reporter.coverage-istanbul]: Coverage for branches (58.33%) does not meet global threshold (80%)
05 05 2017 06:03:47.727:ERROR [reporter.coverage-istanbul]: Coverage for functions (58.33%) does not meet global threshold (80%)
05 05 2017 06:03:47.727:ERROR [reporter.coverage-istanbul]: Coverage for statements (73.33%) in file /Users/yfain11/Downloads/handouts/3-day/A4/unit8/test-weather/src/weather/weather.ts does not meet per file threshold (80%)
05 05 2017 06:03:47.727:ERROR [reporter.coverage-istanbul]: Coverage for lines (69.23%) in file /Users/yfain11/Downloads/handouts/3-day/A4/unit8/test-weather/src/weather/weather.ts does not meet per file threshold (80%)
05 05 2017 06:03:47.727:ERROR [reporter.coverage-istanbul]: Coverage for branches (66.67%) in file /Users/yfain11/Downloads/handouts/3-day/A4/unit8/test-weather/src/weather/weather.ts does not meet per file threshold (80%)
05 05 2017 06:03:47.727:ERROR [reporter.coverage-istanbul]: Coverage for functions (20%) in file /Users/yfain11/Downloads/handouts/3-day/A4/unit8/test-weather/src/weather/weather.ts does not meet per file threshold (80%)
05 05 2017 06:03:47.728:ERROR [reporter.coverage-istanbul]: Coverage for branches (55.56%) in file /Users/yfain11/Downloads/handouts/3-day/A4/unit8/test-weather/src/app/weather.service.ts does not meet per file threshold (80%)
```

Chrome extension Augury

- Augury shows the hierarchy of your app components, services, routes, and injectors.
- Adds an additional Augury tab in the Chrome Dev Tools panel
- Monitor the app objects and modify the property values of components during the runtime.

Generating documentation with compodoc

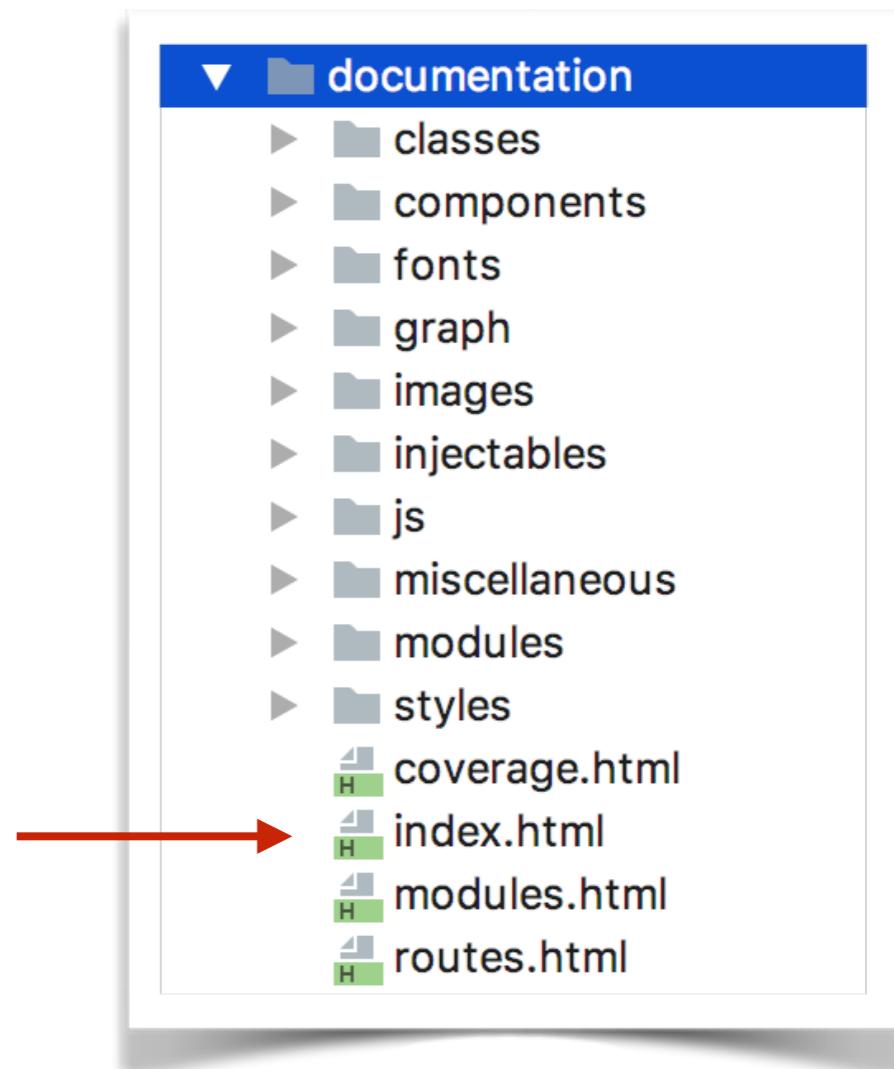
1. Install globally:

```
npm install -g @compodoc/compodoc
```

2. Generate docs in your project:

```
compodoc -p ./tsconfig.json
```

3. Open index.html in your browser



Compodoc-generated docs

ng-auction documentation ↗

Type to search

Getting started

Overview

Modules

Components

AppComponent

CarouselComponent

FooterComponent

HomeComponent

NavbarComponent

ProductDetailComponent

ProductItemComponent

SearchComponent

StarsComponent

AppComponent

AppModule

ProductService

AppRoutingModule

StarsComponent

SearchComponent

ProductDetailComponent

ProductItemComponent

NavbarComponent

FooterComponent

CarouselComponent

HomeComponent

AppCom

Legend

Declarations Module Bootstrap Providers Exports

Zoom in Reset Zoom out

2 modules

9 components

1 injectable

1 routes

1 class

The screenshot shows the Compodoc-generated documentation for a project named "ng-auction". The left sidebar contains a navigation menu with sections for "Getting started", "Overview", "Modules", and "Components". Under "Components", a list of components is shown: AppComponent, CarouselComponent, FooterComponent, HomeComponent, NavbarComponent, ProductDetailComponent, ProductItemComponent, SearchComponent, and StarsComponent. To the right of the sidebar is a large central area. At the top of this area is a diagram illustrating the application's architecture. It shows AppComponent at the top, which is part of the AppModule (indicated by a blue box labeled "Bootstrap"). AppModule also contains declarations for StarsComponent, SearchComponent, ProductDetailComponent, ProductItemComponent, NavbarComponent, FooterComponent, CarouselComponent, and HomeComponent. A legend below the diagram defines symbols: yellow for Declarations, teal for Module, blue for Bootstrap, orange for Providers, and red for Exports. Below the diagram are summary statistics: "2 modules", "9 components", "1 injectable", and "1 routes".

Profile change detection with debug tools

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from './app/app.module';
import { ApplicationRef } from '@angular/core';
import { enableDebugTools } from '@angular/platform-browser';

platformBrowserDynamic().bootstrapModule(AppModule).then((module) => {
  const applicationRef = module.injector.get(ApplicationRef);
  const appComponent = applicationRef.components[0];
  enableDebugTools(appComponent);
});
```

Profile change detection with debug tools

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from './app/app.module';
import { ApplicationRef } from '@angular/core';
import { enableDebugTools } from '@angular/platform-browser';

platformBrowserDynamic().bootstrapModule(AppModule).then((module) => {
  const applicationRef = module.injector.get(ApplicationRef);
  const appComponent = applicationRef.components[0];
  enableDebugTools(appComponent);
});
```

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The output pane displays the results of running the Angular Change Detection profiler.

```
Angular is running in the development mode. Call enableProdMode() to enable the p
> ng.profiler.timeChangeDetection({record: true})
ran 131241 change detection cycles
Profile 'Change Detection' started.
0.00 ms per check
Profile 'Change Detection' finished.
< ► ChangeDetectionPerfRecord {msPerTick: 0.0038121471186595753, numTicks: 131241}
```

The message "Angular is running in the development mode. Call enableProdMode() to enable the p" is highlighted with a red box. The line "ng.profiler.timeChangeDetection({record: true})" is also highlighted with a red box.

ngrx devtools

Monitor the app state changes in the apps that use ngrx

1. Install the Chrome extension Redux
2. `npm install @ngrx/store-devtools`
3. In your app module:

```
StoreDevtoolsModule.instrument()
```

The screenshot shows the NgRx Store DevTools interface. On the left, there's a yellow sidebar with the title "Search component" and a red box containing the text "aaa". Below it, another red box contains the text "eBay Amazon". To the right of the sidebar is a teal sidebar with the title "eBay component" and the text "Search criteria: aaa" followed by a list of five items: "Product aaa0", "Product aaa1", "Product aaa2", "Product aaa3", and "Product aaa4". The main area of the tool has tabs at the top: Elements, Console, Sources, Network, Performance, Memory, Security, Application, Audits, Redux, and a "NgRx Store DevTools" tab which is active. The "Inspector" tab is selected under the "NgRx Store DevTools" tab. On the left side of the inspector, there's a "filter..." input field and a "Commit" button. Below that are four log entries: "@ngrx/store/init" at 10:52:45.14, "@ngrx/effects/init" at 10:52:45.14, "[Product] search" at +00:08.55, and "[Product] search success" at +00:01.00. A red circle with the number "1" is overlaid on the "[Product] search success" entry. On the right side, there's a "State" section with tabs for Tree, Chart, and Raw, with "Tree" selected. It shows a hierarchical state structure: "reducer (pin)" has a child "searchQuery (pin): "aaa"" (red circle with "3" over it) and a child "searchResults (pin)". "searchResults (pin)" has five children labeled 0 through 4, each with the value "Product aaa0", "Product aaa1", "Product aaa2", "Product aaa3", and "Product aaa4" respectively. A red circle with the number "4" is overlaid on the "searchResults" node.

1. The latest action is "[Product] search success"
2. The State tab is selected
3. The search criterion is stored in the state property `searchQuery`
4. The search results are stored in the state property `searchResults`

Search component

aaabbb

[eBay](#) [Amazon](#)

Amazon component

Search criteria:
aaabbb

The screenshot shows the NgRx Store DevTools interface. The top navigation bar includes tabs for Elements, Console, Sources, Network, Performance, Memory, Security, Application, and Redux. The Redux tab is active. Below the tabs is a toolbar with a refresh icon, a pin icon, and five buttons: 'filter...', 'Sweep' (highlighted with a red circle containing '5'), 'Commit', 'Action' (highlighted with a red circle containing '2'), 'State', and 'Diff'. The main area is titled 'Inspector' and shows the 'NgRx Store DevTools' state tree. The 'State' tab is selected. The tree structure under 'myReducer' shows the following state properties:

- searchQuery: "aaabbb" (highlighted with a red circle containing '3')
- searchResults: [] (highlighted with a red circle containing '4')

At the bottom of the tree view, there are 'Jump' and 'Skip' buttons. A red circle containing '1' is positioned at the bottom center of the tree view.

1. We clicked on the Skip button for this action
2. The State tab is selected
3. The search query is aaabbb
4. The state property `searchResults` shows no results for the aaabbb products
5. The Sweep button was not clicked

localhost:4200/#/amazon

Search component

aaa

eBay Amazon

Amazon component

Search criteria: aaa

- Product aaa0
- Product aaa1
- Product aaa2
- Product aaa3
- Product aaa4

Elements Console Sources Network Performance Memory Security Redux

filter... Commit

@ngrx/store/init 6:54:37.08

@ngrx/effects/init 6:54:37.08

ROUTER_NAVIGATION +00:00.16

[Product] search +09:00.71

[Product] search success +00:01.01

ROUTER_NAVIGATION +00:16.74

Inspector NgRx Store DevTools

State Action State Diff

Tree Chart Raw

▶ myReducer (pin): { searchQuery: "aaa", searchResults: [...] , ... }

▼ myRouterReducer (pin)

 ▼ state (pin)

 ▼ _root (pin)

 ▶ value (pin): { url: [/], params: {}, queryParams: {}, ... }

 ▶ children (pin): [...]

 url (pin): "/amazon"

 navigationId (pin): 2

this.

Dispatch

Pause

1x

Linters

Linters help to ensure that the code's written in accepted coding style

TSLint

TSLint is an extensible static analysis tool that checks TypeScript code for readability, maintainability, and functionality errors.

```
tslint --init
```

```
tslint --help
```

```
{  
  "defaultSeverity": "error",  
  "extends": [  
    "tslint:recommended"  
  ],  
  "jsRules": {},  
  "rules": {},  
  "rulesDirectory": []  
}
```

TsLint + codemlyzer

- Angular CLI projects include Codemlyzer.
- Codemlyzer includes the rules conforming to Angular style guide
- To run the linter:
`ng lint`

```
{  
  "rulesDirectory": [  
    "node_modules/codemlyzer"  
  ],  
  "rules": {  
    "callable-types": true,  
    "class-name": true,  
    "comment-format": [  
      true,  
      "check-space"  
    ],  
    "curly": true,  
    "eofline": true,  
    "forin": true,  
    "import-blacklist": [true, "rxjs"],  
    "import-spacing": true,  
    "indent": [  
      true,  
      "spaces"  
    ],  
    "interface-over-type-literal": true,  
    "label-position": true,  
    "max-line-length": [  
      true,  
      140  
    ],  
    "member-access": false,  
    "member-ordering": [  
      true,  
      "static-before-instance",  
      "variables-before-functions"  
    ],  
    "no-arg": true,  
    "no-bitwise": true,  
    "no-console": [  
      true,  
      "debug",  
      "info",  
      "time",  
      "timeEnd",  
      "trace"  
    ]  
  }  
}
```

Thank you!

Blog: yakovfain.com

Twitter: [@yfain](https://twitter.com/@yfain)

email: training@faratasystems.com