

Contents

Sunday, August 6, 2017 9:27 AM

Prerequisites

[Suggested Software](#)

[Recommend npm packages \(for Windows\)](#)

Setup Your Site For Team Activities

[Create Your New Team Camp Site](#)

[Remove The Lint From Your Camp Site](#)

[Support Older Camp Site Browsers](#)

[Refine Scripted Camp Activities](#)

[Bring Along Just The Right Amount Of Style](#)

[Where To Load Stuff That Won't Fit Into Your Pack](#)

[Swap Your Gear When It's Hot](#)

Testing Your Camp Site Modules

[Test Your Gear...Receive Good Karma](#)

[Enhance Your Karma Capabilities](#)

[Make Sure Your Stuff Is Covered](#)

[Your Camp Mascot Is A Wallaby](#)

[Smoke Test Your Camp Site End-to-End](#)

[Form A New Check-in Checklist Habit](#)

Camp Site Construction

[Share Your Camp Tools And Services](#)

[Add Modularized Camp Site Features](#)

[Add Commonly Desired Camp Site Views](#)

[Design The Layout Of Your Camp Site](#)

Add Some Nice-To-Have Amenities

[Feeling Lazy? Load Those Features On Demand](#)

[Name Your Camp Site Locations](#)

[Update Your Current Camp Site Location](#)

[Provide Directives To Guide Campers](#)

[Let Everyone Know When You're Busy](#)

Don't Have All Your Camp Site Supplies Yet? Mock Them Up

[Generate Mock Data](#)

[Setup A JSON Server](#)

[Use Fake Data Generator Libraries](#)

[Use Freely Available Services](#)

Optimize Your Camp Site Resources

[Reduce The Size Of Your Images](#)

[Sprites Reduce Site Downloads](#)

[Bring Along The High Quality Lightweight Gear](#)

[Minimize Your Site Resources](#)

[Build Your Own Custom Font Sets](#)

Extra Camp Site Tasks

[Git Your Camp Site Deployed](#)

[Generate Your FavIcon](#)

[Change The Angular Application Prefix](#)

[Monitor Camp Site Usage And Report Issues](#)

[A Simple ASP.NET Core Site To Host Your Angular SPA](#)

Other Resources

[Learning Site Links](#)

[Pluralsight Videos](#)

[Shiny New Tools](#)

Suggested Software

Saturday, August 5, 2017 7:25 PM

**Visual Studio 2017
(recommend including C++ to support angular-cli node-sass compiler)**

<https://www.visualstudio.com/thank-you-downloading-visual-studio/?sku=Community&rel=15#>

Visual Studio Code (great IDE for JavaScript)

https://code.visualstudio.com/?wt.mc_id=vscom_downloads

Git

<https://git-scm.com/download/win>

NodeJS

<https://nodejs.org/en/>

Yarn

<https://yarnpkg.com/en/>

Wallaby.js

<https://github.com/wallabyjs/ngCliWebpackSample>

Recommend Npm and Yarn Packages (for Windows)

NPM

<https://www.npmjs.com/>

npm install -g minimatch@latest

npm install -g graceful-fs@latest

npm install -g rimraf@latest

npm install -g typescript@latest

npm install -g tslint@latest

Yarn

<https://yarnpkg.com/en/>

yarn config set cache-folder c:\yarn

yarn global add @angular/cli@latest

ng set --global packageManager=yarn

Create Your New Team Camp Site

Thursday, July 20, 2017 6:25 AM

Make Chrome your default browser.

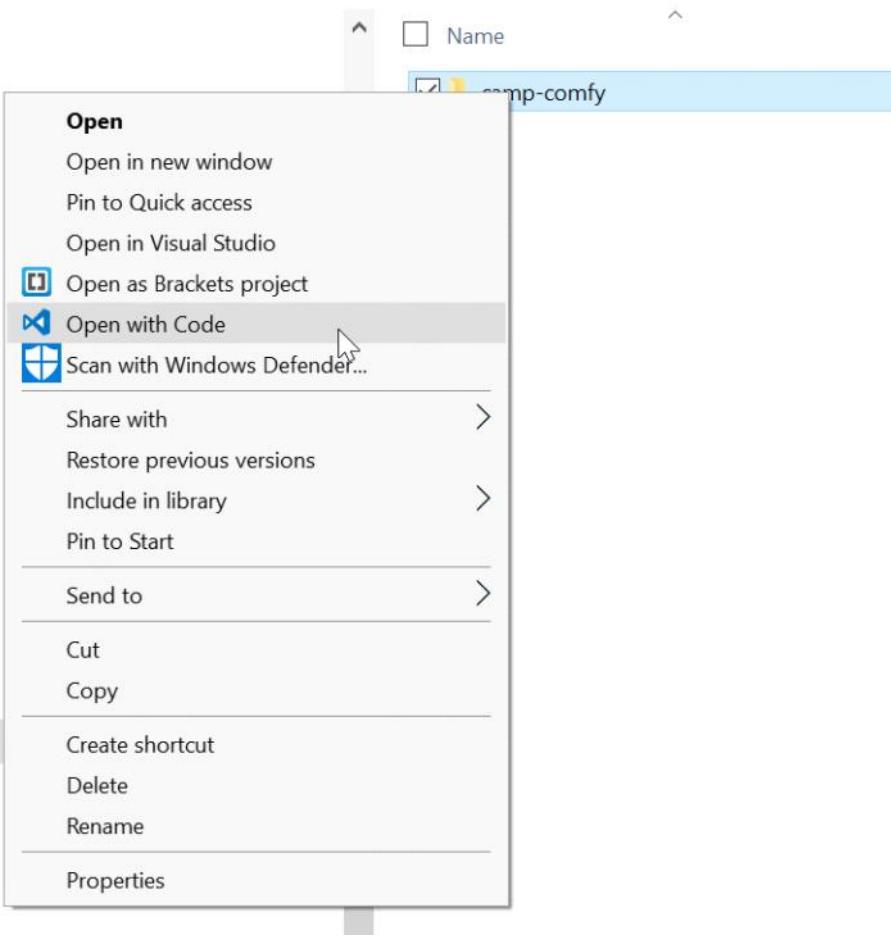
Create a new angular application

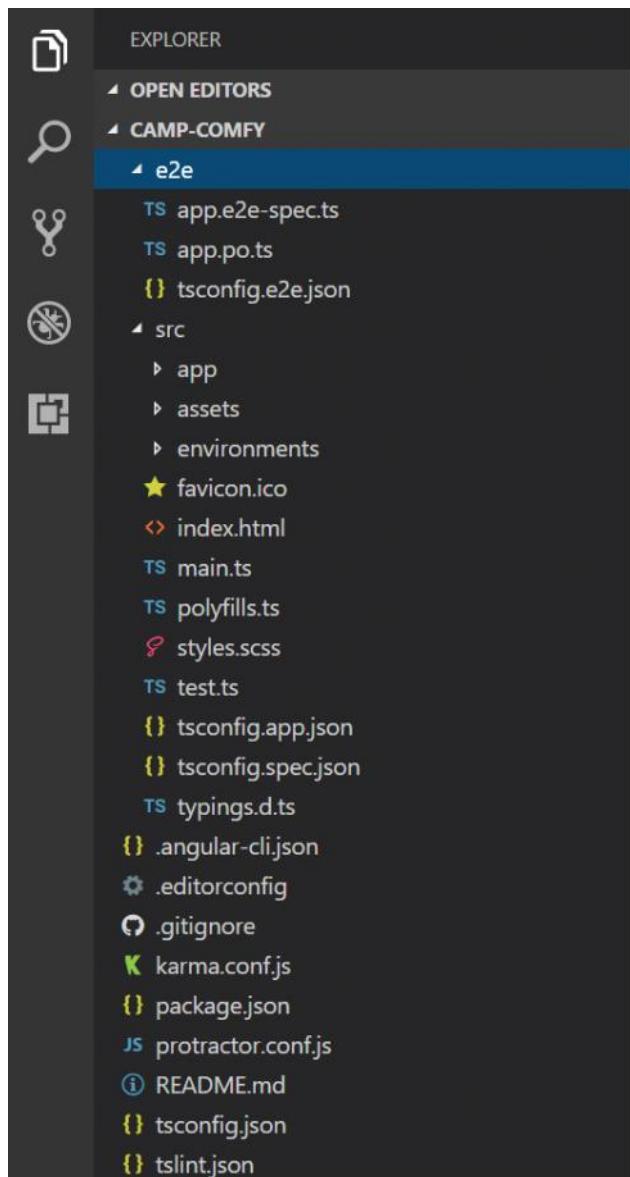
```
ng new camp-comfy --routing --style=scss --prefix=cc
```

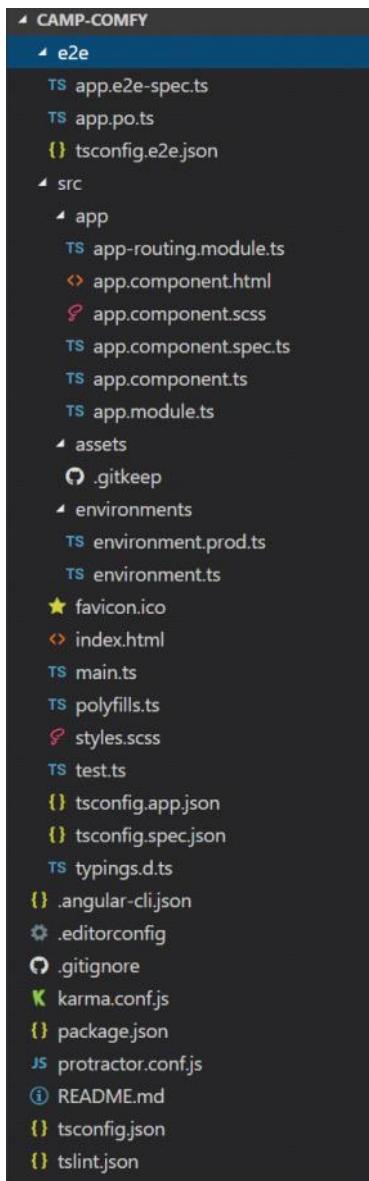
```
PS C:\srcCC> ng new cc-test --routing --style=scss --prefix=cc
installing ng
  create .editorconfig
  create README.md
  create src\app\app-routing.module.ts
  create src\app\app.component.scss
  create src\app\app.component.html
  create src\app\app.component.spec.ts
  create src\app\app.component.ts
  create src\app\app.module.ts
  create src\assets\.gitkeep
  create src\environments\environment.prod.ts
  create src\environments\environment.ts
  create src\favicon.ico
  create src\index.html
  create src\main.ts
  create src\polyfills.ts
  create src\styles.scss
  create src\test.ts
  create src\tsconfig.app.json
  create src\tsconfig.spec.json
  create src\typings.d.ts
  create .angular-cli.json
  create e2e\app.e2e-spec.ts
  create e2e\app.po.ts
  create e2e\tsconfig.e2e.json
  create .gitignore
  create karma.conf.js
  create package.json
  create protractor.conf.js
  create tsconfig.json
  create tslint.json
Installing packages for tooling via npm.
Installed packages for tooling via npm.
Successfully initialized git.
Project 'cc-test' successfully created.
PS C:\srcCC>
```

This may take a few minutes to complete

[Open With Visual Studio Code](#)







The angular-cli tool generates the following:

- Typescript configurations
- Tslint configuration
- Karma configuration
- Protractor configuration
- package.json configuration
- Starter application source code
- Starter unit test source code and Karma configuration
- Starter end-to-end test source code and Protractor configuration
- Git repository

For more detailed information on all of the initially included files, see the following:

What's in the src folder?

<https://angular.io/guide/quickstart#the-src-folder>

What's in the root folder?

<https://angular.io/guide/quickstart#the-root-folder>

Sample packages.json file

```
{  
  "name": "camp-comfy",  
  "version": "0.0.0",  
  "license": "MIT",  
  "scripts": {  
    "ngver": "ng -v",  
    "start": "ng serve --open",  
    "start:prod": "ng serve --prod --open",  
    "build": "ng build",  
    "build:prod": "ng build --prod",  
    "build:prod:folder": "ng build --prod --base-href=/my-app/",  
    "test": "ng test",  
    "lint": "ng lint --type-check --format stylish",  
    "lint:fix": "ng lint --type-check --fix --format stylish",  
    "lint:ci": "ng lint --type-check",  
    "e2e": "ng e2e"  
  },  
  "private": true,  
  "dependencies": {  
    "@angular/animations": "^4.2.4",  
    "@angular/common": "^4.2.4",  
    "@angular/compiler": "^4.2.4",  
    "@angular/core": "^4.2.4",  
    "@angular/forms": "^4.2.4",  
    "@angular/http": "^4.2.4",  
    "@angular/platform-browser": "^4.2.4",  
    "@angular/platform-browser-dynamic": "^4.2.4",  
    "@angular/router": "^4.2.4",  
    "bootstrap": "^4.0.0-beta",  
    "core-js": "^2.4.1",  
    "jquery": "^3.2.1",  
    "popper.js": "^1.12.5",  
    "rxjs": "^5.4.2",  
    "zone.js": "^0.8.14"  
  },  
  "devDependencies": {  
    "@angular/cli": "1.4.2",  
    "@angular/compiler-cli": "^4.2.4",  
    "@angular/language-service": "^4.2.4",  
    "@types/jasmine": "~2.5.53",  
    "@types/jasminewd2": "~2.0.2",  
    "@types/node": "~6.0.60",  
    "angular2-template-loader": "^0.6.2",  
    "codelyzer": "~3.1.1",  
    "jasmine-core": "~2.6.2",  
    "jasmine-spec-reporter": "~4.1.0",  
    "karma": "~1.7.0",  
    "karma-chrome-launcher": "~2.1.1",  
    "karma-cli": "~1.0.1",  
    "karma-coverage-istanbul-reporter": "^1.2.1",  
    "karma-jasmine": "~1.1.0",  
    "karma-jasmine-html-reporter": "^0.2.2",  
    "protractor": "~5.1.2",  
    "ts-node": "~3.2.0",  
    "tslint": "~5.3.2",  
    "typescript": "~2.3.3",  
    "wallaby-webpack": "^0.0.40"  
  }  
}
```

Install the JavaScript Packages

Install the npm packages from the command line:

```
npm install
```

```
C:\srcCC\camp-comfy> npm install
```

```
PS C:\srcCC\camp-comfy> npm install
> node-sass@4.5.3 install c:\srcCC\camp-comfy\node_modules\node-sass
> node scripts/install.js

Cached binary found at c:\users\dmoon\appdata\roaming\npm-cache\node-sass\4.5.3\win32-x64-57_binding.node
> node-sass@4.5.3 postinstall c:\srcCC\camp-comfy\node_modules\node-sass
> node scripts/build.js

Binary found at C:\srcCC\camp-comfy\node_modules\node-sass\vendor\win32-x64-57\binding.node
Testing binary
Binary is fine
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN           SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules\fsevents):
npm WARN           SKIPPING OPTIONAL DEPENDENCY: unsupported platform for fsevents@1.1.2: wanted {"os": "darwin", "arch": "any"} (current: {"os": "win32", "arch": "x64"})
added 1038 packages in 67.347s
PS C:\srcCC\camp-comfy> _
```

Launch app in Chrome

Launch app in IE11, which shows nothing.

Show dev tools console log error

Show initial scripts section in the package.json file:

```
5   "scripts": {
6     "ng": "ng",
7     "start": "ng serve",
8     "build": "ng build",
9     "test": "ng test",
10    "lint": "ng lint",
11    "e2e": "ng e2e"
12  },
```

Start the app using the included lightweight server.

```
npm start
```

```
C:\srcCC\camp-comfy>npm start

> camp-comfy@0.0.0 start C:\srcCC\camp-comfy
> ng serve

** NG Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200 **
Hash: 1253290cae07bda13d3d
Time: 10385ms
chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 177 kB {4} [initial] [rendered]
chunk {1} main.bundle.js, main.bundle.js.map (main) 6.57 kB {3} [initial] [rendered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 10.7 kB {4} [initial] [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.48 MB [initial] [rendered]
chunk {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]
webpack: Compiled successfully.

-
```

Open a browser and enter <http://localhost:4200>

Show the starter tests.

Run built-in tests

```
C:\srcCC\camp-comfy>npm run test

> camp-comfy@0.0.0 test C:\srcCC\camp-comfy
> ng test

10% building modules 1/1 modules 0 active20 07 2017 07:56:53.574:WARN [karma]: No captured browser, open http://localhost:9876/
20 07 2017 07:56:53.590:INFO [karma]: Karma v1.7.0 server started at http://0.0.0.0:9876/
20 07 2017 07:56:53.590:INFO [launcher]: Launching browser Chrome with unlimited concurrency
20 07 2017 07:56:53.598:INFO [launcher]: Starting browser Chrome
20 07 2017 07:57:02.315:WARN [karma]: No captured browser, open http://localhost:9876/
20 07 2017 07:57:02.796:INFO [Chrome 59.0.3071 (Windows 10 0.0.0)]: Connected on socket v90GHjykFCLgT4RYAAAA with id 27220391
Chrome 59.0.3071 (Windows 10 0.0.0): Executed 3 of 3 SUCCESS (0.273 secs / 0.257 secs)
-
```

Karma v1.7.0 - connected

DEBUG

Chrome 59.0.3071 (Windows 10 0.0.0) is idle

Jasmine 2.6.4

finished in 0.267s

• • •

3 specs, 0 failures

raise exceptions

```
AppComponent
  should create the app
  should have as title 'app'
  should render title in a h1 tag
```

Welcome to app!



Here are some links to help you start:

- [Tour of Heroes](#)

Show the starter e2e test

Run e2e tests.

```
C:\srcCC\camp-comfy>npm run e2e
> camp-comfy@0.0.0 e2e C:\srcCC\camp-comfy
> ng e2e

** NG Live Development Server is listening on localhost:49152, open your browser on http://localhost:49152 **
(node:4496) [DEP0022] DeprecationWarning: os.tmpDir() is deprecated. Use os.tmpdir() instead.
Hash: c2f515d784504f8853ef
Time: 10795ms
chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 177 kB {4} [initial] [rendered]
chunk {1} main.bundle.js, main.bundle.js.map (main) 6.57 kB {3} [initial] [rendered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 10.7 kB {4} [initial] [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.48 MB [initial] [rendered]
chunk {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]
webpack: Compiled successfully.
[07:59:46] I/file_manager - creating folder C:\srcCC\camp-comfy\node_modules\protractor\node_modules\webdriver-manager\selenium
[07:59:47] I/update - chromedriver: unzipping chromedriver_2.30.zip
[07:59:47] I/launcher - Running 1 instances of WebDriver
[07:59:47] I/direct - Using ChromeDriver directly...
Jasmine started

  camp-comfy App
    ✓ should display welcome message

Executed 1 of 1 spec SUCCESS in 1 sec.
[07:59:52] I/launcher - 0 instance(s) of WebDriver still running
[07:59:52] I/launcher - chrome #01 passed
```

Ahead of Time (AoT) Compilation

Tree shaking

Tree shaking can greatly reduce the downloaded size of the application by removing unused portions of both source and library code. In fact, most of the reduction in small apps comes from removing

unreferenced Angular features.

AOT compilation sets the stage for further optimization through a process called tree shaking. A tree shaker walks the dependency graph, top to bottom, and shakes out unused code like dead leaves in a tree.

Making your Angular 2 library statically analyzable for AoT

<https://medium.com/@isaacplmann/making-your-angular-2-library-statically-analyzable-for-aot-e1c6f3ebbedd5>

Remove The Lint From Your Camp Site

Friday, July 21, 2017 5:24 PM

Remove a semicolon and the last blank line in the **app.component.ts** file.

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
8 export class AppComponent {
9   title = 'app'
10 }
```

Run the lint script.

```
npm run lint
```

```
c:\srcCC\camp-comfy>npm run lint
> camp-comfy@0.0.0 lint C:\srcCC\camp-comfy
> ng lint --format stylish

Warning: The 'no-use-before-declare' rule requires type checking
C:/srcCC/camp-comfy/src/app/app.component.ts
ERROR: 10:2 eofline    file should end with a newline
ERROR: 9:16 semicolon  Missing semicolon

Lint errors found in the listed files.
npm ERR! code ELIFECYCLE
npm ERR! errno 2
npm ERR! camp-comfy@0.0.0 lint: `ng lint --format stylish`
npm ERR! Exit status 2
npm ERR!
npm ERR! Failed at the camp-comfy@0.0.0 lint script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!     C:\Users\dmoon\AppData\Roaming\npm-cache\_logs\2017-07-20T13_29_18_847Z-debug.log
```

Notice that a couple of errors were reported by tslint.

Run the **lint:fix** script (without the **--no-type-check** argument).

```
npm run lint:fix
```

```
C:\srcCC\camp-comfy>npm run lint:fix

> camp-comfy@0.0.0 lint:fix C:\srcCC\camp-comfy
> ng lint --fix --format stylish

Warning: The 'no-use-before-declare' rule requires type checking

All files pass linting.

C:\srcCC\camp-comfy>
```

Notice that the linting errors are gone, but you still see the warning 'no-use-before-declare'.

Add the --type-check argument to script.

Run the lint:fix script (with the --type-check argument).

npm run lint:fix

```
5   "scripts": {
6     "ngver": "ng -v",
7     "start": "ng serve --open",
8     "start:prod": "ng serve --prod --open",
9     "build": "ng build",
10    "build:prod": "ng build --prod",
11    "build:prod:folder": "ng build --prod --base-href=/my-app/",
12    "test": "ng test",
13    "lint": "ng lint --type-check --format stylish",
14    "lint:fix": "ng lint --type-check --fix --format stylish",
15    "lint:ci": "ng lint --type-check",
16    "e2e": "ng e2e"
17  },
```

Run the lint-fix script again.

```
All files pass linting.

C:\srcCC\camp-comfy>npm run lint:fix

> camp-comfy@0.0.0 lint:fix C:\srcCC\camp-comfy
> ng lint --type-check --fix --format stylish

All files pass linting.
```

All file pass linting.

Look again at the **app.component.ts** file.

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
8 export class AppComponent {
9   title = 'app';
10 }
11 
```

Notice that the semicolon on the title statement has returned on line 9, and an ending linefeed now exists on line 11.

Support Older Camp Site Browsers

Thursday, July 20, 2017 6:56 AM

Add Older Browser Polyfills

Run the start script.

```
npm run start
```

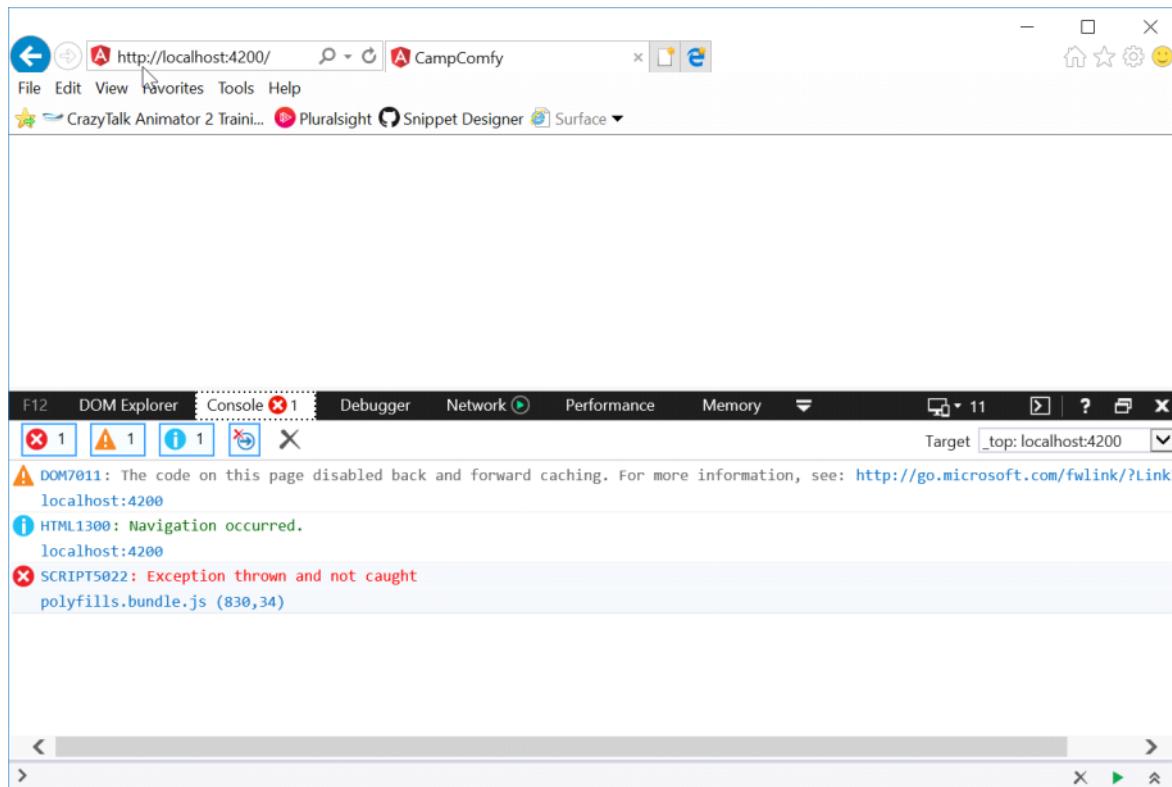
Launch IE11.

Open the developer tools.

Select the debugging console.

Navigate to <http://localhost:4200>

You should see the following error.



This indicates that the older version of IE does not have the required support for this version of Angular.

However, we can remedy this by installing some shims (polyfills) to add the missing capabilities.

NOTE: That the latest version of Angular is not fully supported on IE browsers older than IE9.

From within the src folder, open the file called polyfills.ts

This file includes polyfills needed by Angular and is loaded before the app.
You can add your own extra polyfills to this file.

This file is divided into 2 sections:

1. Browser polyfills. These are applied before loading ZoneJS and are sorted by browsers.
2. Application imports. Files imported after ZoneJS that should be loaded before your main file.

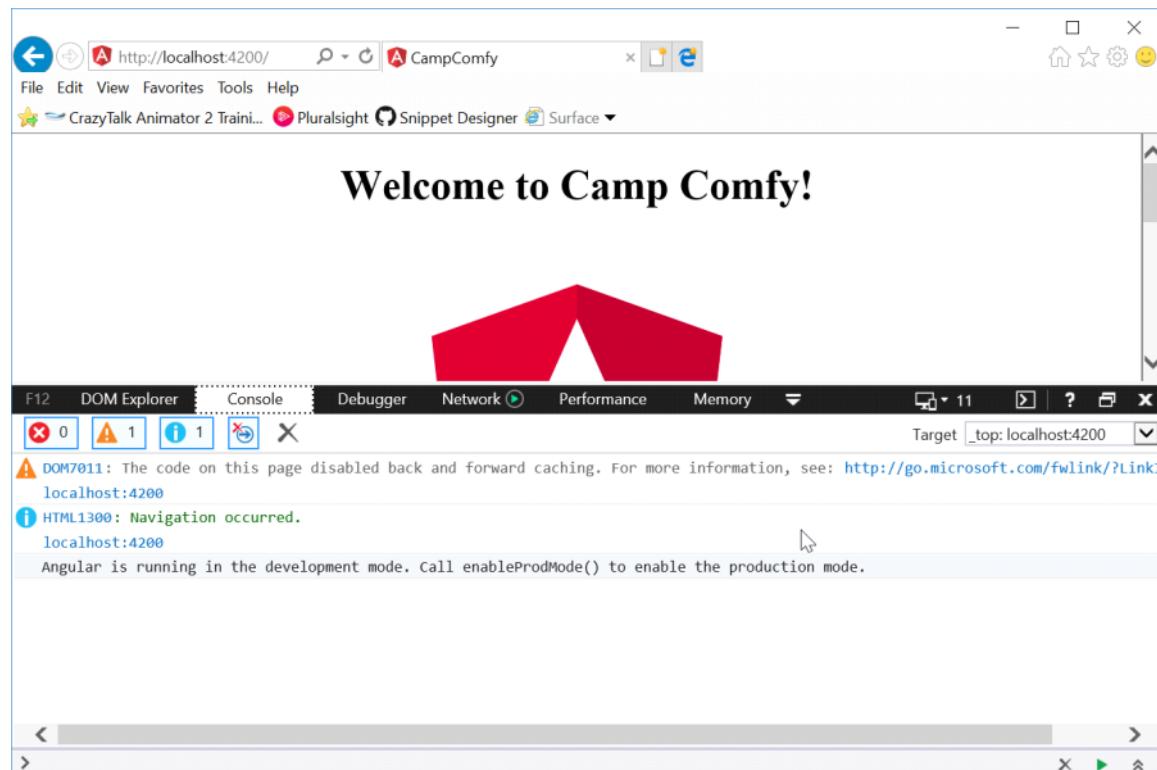
The current setup is for so-called "evergreen" browsers; the last versions of browsers that automatically update themselves. This includes Safari >= 10, Chrome >= 55 (including Opera), Edge >= 13 on the desktop, and iOS 10 and Chrome on mobile.

Learn more in <https://angular.io/docs/ts/latest/guide/browser-support.html>

Uncomment the following browser polyfills and save the file.

```
/** IE9, IE10 and IE11 requires all of the following polyfills. **/
import 'core-js/es6/symbol';
import 'core-js/es6/object';
import 'core-js/es6/function';
import 'core-js/es6/parse-int';
import 'core-js/es6/parse-float';
import 'core-js/es6/number';
import 'core-js/es6/math';
import 'core-js/es6/string';
import 'core-js/es6/date';
import 'core-js/es6/array';
import 'core-js/es6/regexp';
import 'core-js/es6/map';
import 'core-js/es6/weak-map';
import 'core-js/es6/set';
```

Show how this resolves the IE11 compatibility issue.



Note that the minimized polyfills went from 62KB

3rdpartylicenses.txt	7/20/2017 10:15 AM	Text Document	0 KB
favicon.ico	7/20/2017 10:15 AM	Icon	6 KB
index.html	7/20/2017 10:15 AM	Chrome HTML Docu...	1 KB
inline.18a0b543ad103a3ec7a3.bundle.js	7/20/2017 10:15 AM	JavaScript File	2 KB
main.8f49e9573568b29c8697.bundle.js	7/20/2017 10:15 AM	JavaScript File	7 KB
polyfills.1553fdd8173a081c720c.bundle.js	7/20/2017 10:15 AM	JavaScript File	62 KB
styles.d41d8cd98f00b204e980.bundle.css	7/20/2017 10:15 AM	Cascading Style Shee...	0 KB
vendor.db8b86c0f30bd419d13d.bundle.js	7/20/2017 10:15 AM	JavaScript File	299 KB

To 95KB (+33KB)

3rdpartylicenses.txt	7/20/2017 10:22 AM	Text Document	0 KB
favicon.ico	7/20/2017 10:22 AM	Icon	6 KB
index.html	7/20/2017 10:22 AM	Chrome HTML Docu...	1 KB
inline.11613c03bd6d2e4cd6d6.bundle.js	7/20/2017 10:22 AM	JavaScript File	2 KB
main.8f49e9573568b29c8697.bundle.js	7/20/2017 10:22 AM	JavaScript File	7 KB
polyfills.c7f2b30f4d8ff16ddd31.bundle.js	7/20/2017 10:22 AM	JavaScript File	95 KB
styles.d41d8cd98f00b204e980.bundle.css	7/20/2017 10:22 AM	Cascading Style Shee...	0 KB
vendor.db8b86c0f30bd419d13d.bundle.js	7/20/2017 10:22 AM	JavaScript File	299 KB

Additional Browser PolyFills

NOTE: Only install the following if you need the indicated angular capability.

Add Support For SVG Graphics

Run the following package installation:

```
npm install --save classlist.js
```

```
PS C:\srcCC\camp-comfy> npm install --save classlist.js
npm WARN      SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules\fsevents):
npm WARN      SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"}
"{}" (current: {"os":"win32","arch":"x64"})
+ classlist.js@1.1.20150312
added 115 packages in 17.994s
PS C:\srcCC\camp-comfy>
```

Add Support For Animations

Run the following package installation:

```
npm install --save web-animations-js
```

```

PS C:\srcCC\camp-comfy> npm install --save web-animations-js
npm WARN      SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules\fsevents):
npm WARN      SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"}
"{} (current: {"os":"win32","arch":"x64"})
+ web-animations-js@2.3.1
added 115 packages in 17.214s
PS C:\srcCC\camp-comfy> -

```

Application Support For Date, Currency, Decimal And Percent Pipes

Run the following package installation:

```
npm install --save intl
```

```

PS C:\srcCC\camp-comfy> npm install --save intl
npm WARN      SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules\fsevents):
npm WARN      SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"}
"{} (current: {"os":"win32","arch":"x64"})
+ intl@1.2.5
added 115 packages in 18.053s
PS C:\srcCC\camp-comfy>

```

With all polyfills libraries installed, you should now see the following new items in the dependencies section of the package.json file.

```

14   "dependencies": {
15     "@angular/animations": "^4.0.0",
16     "@angular/common": "^4.0.0",
17     "@angular/compiler": "^4.0.0",
18     "@angular/core": "^4.0.0",
19     "@angular/forms": "^4.0.0",
20     "@angular/http": "^4.0.0",
21     "@angular/platform-browser": "^4.0.0",
22     "@angular/platform-browser-dynamic": "^4.0.0",
23     "@angular/router": "^4.0.0",
24 +   "classlist.js": "^1.1.20150312",
25     "core-js": "^2.4.1",
26 +   "intl": "^1.2.5",
27     "rxjs": "^5.1.0",
28 +   "web-animations-js": "^2.3.1",
29     "zone.js": "^0.8.4"
30   },

```

Example polyfills.ts File With Everything Enabled

```

/**
 * This file includes polyfills needed by Angular and is loaded before the app.
 * You can add your own extra polyfills to this file.
 *
 * This file is divided into 2 sections:
 * 1. Browser polyfills. These are applied before loading ZoneJS and are sorted by browsers.
 * 2. Application imports. Files imported after ZoneJS that should be loaded before your main
 * file.
 *
 * The current setup is for so-called "evergreen" browsers; the last versions of browsers that
 * automatically update themselves. This includes Safari >= 10, Chrome >= 55 (including Opera),
 * Edge >= 13 on the desktop, and iOS 10 and Chrome on mobile.

```

```

/*
 * Learn more in https://angular.io/docs/ts/latest/guide/browser-support.html
*/
/
*****
*
* BROWSER POLYFILLS
*/
/** IE9, IE10 and IE11 requires all of the following polyfills. */
import 'core-js/es6/symbol';
import 'core-js/es6/object';
import 'core-js/es6/function';
import 'core-js/es6/parse-int';
import 'core-js/es6/parse-float';
import 'core-js/es6/number';
import 'core-js/es6/math';
import 'core-js/es6/string';
import 'core-js/es6/date';
import 'core-js/es6/array';
import 'core-js/es6/regexp';
import 'core-js/es6/map';
import 'core-js/es6/weak-map';
import 'core-js/es6/set';
/** IE10 and IE11 requires the following for NgClass support on SVG elements */
// import 'classlist.js'; // Run `npm install --save classlist.js`.
/** Evergreen browsers require these. */
import 'core-js/es6/reflect';
import 'core-js/es7/reflect';

/**
 * Required to support Web Animations `@angular/animations`.
 * Needed for: All but Chrome, Firefox and Opera. http://caniuse.com/#feat=web-animation
**/
// import 'web-animations-js'; // Run `npm install --save web-animations-js`.

/
*****
*
* Zone JS is required by Angular itself.
*/
import 'zone.js/dist/zone'; // Included with Angular CLI.

/
*****
*
* APPLICATION IMPORTS
*/
/** 
* Date, currency, decimal and percent pipes.
* Needed for: All but Chrome, Firefox, Edge, IE11 and Safari 10
*/
// import 'intl'; // Run `npm install --save intl`.
/** 
* Need to import at least one locale-data with intl.
*/
// import 'intl/locale-data/jsonp/en';

```

Refine Scripted Camp Activities

Friday, July 21, 2017 5:27 PM

Upgrade The Scripts Block

Edit the **package.json** file.

Replace the scripts block with the following:

```
5   "scripts": {  
6     "ngver": "ng -v",  
7     "start": "ng serve --open",  
8     "start:prod": "ng serve --prod --open",  
9     "build": "ng build",  
10    "build:prod": "ng build --prod",  
11    "build:prod:folder": "ng build --prod --base-href=/my-app/",  
12    "test": "ng test",  
13    "lint": "ng lint --format stylish",  
14    "lint:fix": "ng lint --fix --format stylish",  
15    "lint:ci": "ng lint",  
16    "e2e": "ng e2e"  
17  },
```

```
"scripts": {  
  "ngver": "ng -v",  
  "start": "ng serve --open",  
  "start:prod": "ng serve --prod --open",  
  "build": "ng build",  
  "build:prod": "ng build --prod",  
  "build:prod:folder": "ng build --prod --base-href=/my-app/",  
  "test": "ng test",  
  "lint": "ng lint --type-check --format stylish",  
  "lint:fix": "ng lint --type-check --fix --format stylish",  
  "lint:ci": "ng lint --type-check",  
  "e2e": "ng e2e"  
},
```

Check The Current Angular Version

Run the ngver script.

```
npm run ngver
```

```
C:\srcCC\camp-comfy>npm run ngver
> camp-comfy@0.0.0 ngver C:\srcCC\camp-comfy
> ng -v

Angular CLI: 1.2.1
Node: 8.1.4
OS: win32 x64
Angular: 4.3.1
  @angular/animations: 4.3.1
  @angular/common: 4.3.1
  @angular/compiler: 4.3.1
  @angular/core: 4.3.1
  @angular/forms: 4.3.1
  @angular/http: 4.3.1
  @angular/platform-browser: 4.3.1
  @angular/platform-browser-dynamic: 4.3.1
  @angular/router: 4.3.1
  @angular/cli: 1.2.1
  @angular/compiler-cli: 4.3.1
  @angular/language-service: 4.3.1

C:\srcCC\camp-comfy>
```

You should see the version of angular-cli and the versions of the angular modules in the project.

Do A Development Build

Run the build script.

npm run build

```
C:\srcCC\camp-comfy>npm run build

> camp-comfy@0.0.0 build C:\srcCC\camp-comfy
> ng build

Hash: e1b4b586d7df4d3fd4b4
Time: 9258ms
chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 177 kB {4} [initial] [rendered]
chunk {1} main.bundle.js, main.bundle.js.map (main) 6.56 kB {3} [initial] [rendered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 10.7 kB {4} [initial] [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.18 MB [initial] [rendered]
chunk {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]

C:\srcCC\camp-comfy>
```

 favicon.ico	7/20/2017 10:18 AM	Icon	6 KB
 index.html	7/20/2017 10:18 AM	Chrome HTML Docu...	1 KB
 inline.bundle.js	7/20/2017 10:18 AM	JavaScript File	6 KB
 inline.bundle.js.map	7/20/2017 10:18 AM	Linker Address Map	6 KB
 main.bundle.js	7/20/2017 10:18 AM	JavaScript File	11 KB
 main.bundle.js.map	7/20/2017 10:18 AM	Linker Address Map	7 KB
 polyfills.bundle.js	7/20/2017 10:18 AM	JavaScript File	198 KB
 polyfills.bundle.js.map	7/20/2017 10:18 AM	Linker Address Map	228 KB
 styles.bundle.js	7/20/2017 10:18 AM	JavaScript File	12 KB
 styles.bundle.js.map	7/20/2017 10:18 AM	Linker Address Map	15 KB
 vendor.bundle.js	7/20/2017 10:18 AM	JavaScript File	2,208 KB
 vendor.bundle.js.map	7/20/2017 10:18 AM	Linker Address Map	2,689 KB

Note the creation of map files for debugging use and also note the size of the files.

Do A Production Build

Run the build:prod script.

npm run build:prod

```
C:\srcCC\camp-comfy>npm run build:prod

> camp-comfy@0.0.0 build:prod C:\srcCC\camp-comfy
> ng build --prod

Hash: c3983e096367cf21abc2
Time: 11485ms
chunk {0} polyfills.1553fdd8173a081c720c.bundle.js (polyfills) 177 kB {4} [initial] [rendered]
chunk {1} main.5f27b22cde1d1b1b24a5.bundle.js (main) 16.2 kB {3} [initial] [rendered]
chunk {2} styles.d41d8cd98f00b204e980.bundle.css (styles) 69 bytes {4} [initial] [rendered]
chunk {3} vendor.db8b86c0f30bd419d13d.bundle.js (vendor) 1.14 MB [initial] [rendered]
chunk {4} inline.1415c2ea1dea0776629f.bundle.js (inline) 0 bytes [entry] [rendered]

C:\srcCC\camp-comfy>
```

 3rdpartylicenses.txt	7/20/2017 10:15 AM	Text Document	0 KB
 favicon.ico	7/20/2017 10:15 AM	Icon	6 KB
 index.html	7/20/2017 10:15 AM	Chrome HTML Docu...	1 KB
 inline.18a0b543ad103a3ec7a3.bundle.js	7/20/2017 10:15 AM	JavaScript File	2 KB
 main.8f49e9573568b29c8697.bundle.js	7/20/2017 10:15 AM	JavaScript File	7 KB
 polyfills.1553fdd8173a081c720c.bundle.js	7/20/2017 10:15 AM	JavaScript File	62 KB
 styles.d41d8cd98f00b204e980.bundle.css	7/20/2017 10:15 AM	Cascading Style Shee...	0 KB
 vendor.db8b86c0f30bd419d13d.bundle.js	7/20/2017 10:15 AM	JavaScript File	299 KB

Note the absence of map files and also observe the greatly reduces size of the files.

Run The Start Task

Start the app using the enhanced start script.

```
npm run start
```

```
C:\srcCC\camp-comfy>npm run start
> camp-comfy@0.0.0 start C:\srcCC\camp-comfy
> ng serve --open

** NG Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200 **
10% building modules 6/9 modules 3 active ...ode_modules\style-loader\addStyles.jswebpack: wait until bundle finished:
/                                         Hash: 1253290cae07bda13d3d
Time: 12128ms
chunk  {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 177 kB {4} [initial] [rendered]
chunk  {1} main.bundle.js, main.bundle.js.map (main) 6.57 kB {3} [initial] [rendered]
chunk  {2} styles.bundle.js, styles.bundle.js.map (styles) 10.7 kB {4} [initial] [rendered]
chunk  {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.48 MB [initial] [rendered]
chunk  {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]
webpack: Compiled successfully.
```

The angular lightweight web server automatically launches the application in your default browser.

Welcome to app!



Now change the title in the app.component.ts file, quickly switch to the browser and watch it refresh with the updated title. "Camp Comfy"

Welcome to Camp Comfy!



Bring Along Just The Right Amount Of Style

Thursday, July 20, 2017 9:49 AM

Install Bootstrap 4

Add the next version of bootstrap to the project.

npm install bootstrap@next

Or

yarn add bootstrap@next

Add Local Bootstrap Override Files

In the src folder, create a new file called:

_variables.scss

Add nothing to this file at this time.

In the src folder, create a new file called:

_bootstrap.scss

Add the following content by copying it from the bootstrap.scss source file in the node_module/bootstrap/scss folder, and alter the import statement to include the paths to the source files:

```
/*
 * Bootstrap v4.0.0-beta (https://getbootstrap.com)
 * Copyright 2011-2017 The Bootstrap Authors
 * Copyright 2011-2017 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */
@import "../node_modules/bootstrap/scss/functions";
@import "../node_modules/bootstrap/scss/variables";
@import "../node_modules/bootstrap/scss/mixins";
@import "../node_modules/bootstrap/scss/print";
@import "../node_modules/bootstrap/scss/reboot";
@import "../node_modules/bootstrap/scss/type";
@import "../node_modules/bootstrap/scss/images";
@import "../node_modules/bootstrap/scss/code";
@import "../node_modules/bootstrap/scss/grid";
@import "../node_modules/bootstrap/scss/tables";
@import "../node_modules/bootstrap/scss/forms";
@import "../node_modules/bootstrap/scss/buttons";
@import "../node_modules/bootstrap/scss/transitions";
@import "../node_modules/bootstrap/scss/dropdown";
@import "../node_modules/bootstrap/scss/button-group";
@import "../node_modules/bootstrap/scss/input-group";
```

```
@import "../node_modules/bootstrap/scss/custom-forms";
@import "../node_modules/bootstrap/scss/nav";
@import "../node_modules/bootstrap/scss/navbar";
@import "../node_modules/bootstrap/scss/card";
@import "../node_modules/bootstrap/scss/breadcrumb";
@import "../node_modules/bootstrap/scss/pagination";
@import "../node_modules/bootstrap/scss/badge";
@import "../node_modules/bootstrap/scss/jumbotron";
@import "../node_modules/bootstrap/scss/alert";
@import "../node_modules/bootstrap/scss/progress";
@import "../node_modules/bootstrap/scss/media";
@import "../node_modules/bootstrap/scss/list-group";
@import "../node_modules/bootstrap/scss/close";
@import "../node_modules/bootstrap/scss/modal";
@import "../node_modules/bootstrap/scss/tooltip";
@import "../node_modules/bootstrap/scss/popover";
@import "../node_modules/bootstrap/scss/carousel";
@import "../node_modules/bootstrap/scss/utilities";
```

In the styles.scss file, add the following:

```
@import 'variables';
@import 'bootstrap';
```

These point to the _variables.scss and _bootstrap.scss file you created in the src folder.

Test The New CSS

Open a command line windows and run the start script.

```
npm run start
```

The site should now appear using a sans-serif font:

Welcome to Camp Comfy!



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

Stop the site.

Optimize The Size Of The Styles File

Open a new command line window.

Run the build:prod script.

npm run build:prod

Observe that the styles file size when from 0KB:

	3rdpartylicenses.txt	7/20/2017 10:22 AM	Text Document	0 KB
	favicon.ico	7/20/2017 10:22 AM	Icon	6 KB
	index.html	7/20/2017 10:22 AM	Chrome HTML Docu...	1 KB
	inline.11613c03bd6d2e4cd6d6.bundle.js	7/20/2017 10:22 AM	JavaScript File	2 KB
	main.8f49e9573568b29c8697.bundle.js	7/20/2017 10:22 AM	JavaScript File	7 KB
	polyfills.c7f2b30f4d8ff16ddd31.bundle.js	7/20/2017 10:22 AM	JavaScript File	95 KB
	styles.d41d8cd98f00b204e980.bundle.css	7/20/2017 10:22 AM	Cascading Style Shee...	0 KB
	vendor.db8b86c0f30bd419d13d.bundle.js	7/20/2017 10:22 AM	JavaScript File	299 KB

In the **styles.scss** file, add the following and save the file:

```
@import 'variables';
@import 'bootstrap';
```

These point to the `_variables.scss` and `_bootstrap.scss` file you created in the `src` folder.

Run the `build:prod` script.

```
npm run build:prod
```

Observe that the styles file size is now 132KB (+132KB)

 3rdpartylicenses.txt	7/20/2017 10:24 AM	Text Document	0 KB
 favicon.ico	7/20/2017 10:24 AM	Icon	6 KB
 index.html	7/20/2017 10:24 AM	Chrome HTML Docu...	1 KB
 inline.11613c03bd6d2e4cd6d6.bundle.js	7/20/2017 10:24 AM	JavaScript File	2 KB
 main.8f49e9573568b29c8697.bundle.js	7/20/2017 10:24 AM	JavaScript File	7 KB
 polyfills.c7f2b30f4d8ff16ddd31.bundle.js	7/20/2017 10:24 AM	JavaScript File	95 KB
 styles.98cc7080182c73a0ce69.bundle.css	7/20/2017 10:24 AM	Cascading Style Shee...	132 KB
 vendor.db8b86c0f30bd419d13d.bundle.js	7/20/2017 10:24 AM	JavaScript File	299 KB

You should run with all of the bootstrap styles until you near the end of your development cycle.

Optimize Your Styles CSS File Size

Once you near the end of your development, you can optimize the bootstrap stylesheet by commenting out unused styles in the `_bootstrap.scss` file.

Comment out the following styles in the `_bootstrap.scss` file:

```
// Components
@import "../node_modules/bootstrap/scss/transitions";
@import "../node_modules/bootstrap/scss/dropdown";
// @import "../node_modules/bootstrap/scss/button-group";
@import "../node_modules/bootstrap/scss/input-group";
@import "../node_modules/bootstrap/scss/custom-forms";
@import "../node_modules/bootstrap/scss/nav";
@import "../node_modules/bootstrap/scss/navbar";
// @import "../node_modules/bootstrap/scss/card";
// @import "../node_modules/bootstrap/scss/breadcrumb";
// @import "../node_modules/bootstrap/scss/pagination";
// @import "../node_modules/bootstrap/scss/badge";
// @import "../node_modules/bootstrap/scss/jumbotron";
@import "../node_modules/bootstrap/scss/alert";
// @import "../node_modules/bootstrap/scss/progress";
@import "../node_modules/bootstrap/scss/media";
// @import "../node_modules/bootstrap/scss/list-group";
@import "../node_modules/bootstrap/scss/responsive-embed";
@import "../node_modules/bootstrap/scss/close";
```

```
// Components w/ JavaScript
@import "../node_modules/bootstrap/scss/modal";
// @import "../node_modules/bootstrap/scss/tooltip";
// @import "../node_modules/bootstrap/scss/popover";
// @import "../node_modules/bootstrap/scss/carousel";
```

Run the build:prod script.

npm run build:prod

Observe that the styles file size went from 132KB:

 3rdpartylicenses.txt	7/20/2017 10:24 AM	Text Document	0 KB
 favicon.ico	7/20/2017 10:24 AM	Icon	6 KB
 index.html	7/20/2017 10:24 AM	Chrome HTML Docu...	1 KB
 inline.11613c03bd6d2e4cd6d6.bundle.js	7/20/2017 10:24 AM	JavaScript File	2 KB
 main.8f49e9573568b29c8697.bundle.js	7/20/2017 10:24 AM	JavaScript File	7 KB
 polyfills.c7f2b30f4d8ff16ddd31.bundle.js	7/20/2017 10:24 AM	JavaScript File	95 KB
 styles.98cc7080182c73a0ce69.bundle.css	7/20/2017 10:24 AM	Cascading Style Shee...	132 KB
 vendor.db8b86c0f30bd419d13d.bundle.js	7/20/2017 10:24 AM	JavaScript File	299 KB

To 107KB (-25KB)

 3rdpartylicenses.txt	7/20/2017 10:32 AM	Text Document	0 KB
 favicon.ico	7/20/2017 10:32 AM	Icon	6 KB
 index.html	7/20/2017 10:32 AM	Chrome HTML Docu...	1 KB
 inline.11613c03bd6d2e4cd6d6.bundle.js	7/20/2017 10:32 AM	JavaScript File	2 KB
 main.8f49e9573568b29c8697.bundle.js	7/20/2017 10:32 AM	JavaScript File	7 KB
 polyfills.c7f2b30f4d8ff16ddd31.bundle.js	7/20/2017 10:32 AM	JavaScript File	95 KB
 styles.6dd9273f3654f3483216.bundle.css	7/20/2017 10:32 AM	Cascading Style Shee...	107 KB
 vendor.db8b86c0f30bd419d13d.bundle.js	7/20/2017 10:32 AM	JavaScript File	299 KB

Every little bit of size reduction helps reduce application load times.

Testing The Project

Open **app.component.html** and add the following markup:

```
<button class="btn btn-primary">Test Button</button>
```

```
1  <!--The content below is only a placeholder and can be replaced.-->
2  <div style="text-align:center">
3    <h1>
4      | Welcome to {{title}}!
5    </h1>
6    Tour of Heroes</a></h2>
12   </li>
13   <li>
14     | <h2><a target="_blank" href="https://github.com/angular/angular-cli/wiki">CLI Documentation</a></h2>
15   </li>
16   <li>
17     | <h2><a target="_blank" href="http://angularjs.blogspot.ca/">Angular blog</a></h2>
18   </li>
19 </ul>
20
21 <button class="btn btn-primary">Test Button</button>
22
23 <router-outlet></router-outlet>
24 |
```

Edit the `_variables.scss` file and add the following:

```
$brand-primary: red;
```

The app should now show the primary red color and the new bootstrap button:

Welcome to Camp Comfy!



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

Test Button



See the following angular-cli documentation for more information:

<https://github.com/angular/angular-cli/wiki/stories-include-bootstrap>

Where To Load Stuff That Won't Fit Into Your Pack

Thursday, July 20, 2017 10:42 AM

Install your 3rd party library, as follows:

npm install library-name --save

Or

yarn add library-name

and import it in your code.

If the library does not include typings, install them using npm:

npm install @types/library_name --save-dev

Or

yarn add @types/library_name --dev

As an example, if you wish to use some of the bootstrap controls that require additional JavaScript dependencies that are not written using the CommonJS standard, you can add references to them in the "scripts" section in the **.angular-cli.json** file, as follows:

```
21      "styles": [
22        "styles.scss"
23      ],
24      "scripts": [
25        "../node_modules/jquery/dist/jquery.slim.min.js",
26        "../node_modules/tether/dist/js/tether.min.js",
27        "../node_modules/bootstrap/dist/js/bootstrap.min.js"
28    ],
```

For bootstrap JavaScript, you could optimize even further by loading individual control scripts.

Observe original files before adding the scripts to the **.angular-cli.json** file.

 3rdpartylicenses.txt	7/20/2017 10:32 AM	Text Document	0 KB
 favicon.ico	7/20/2017 10:32 AM	Icon	6 KB
 index.html	7/20/2017 10:32 AM	Chrome HTML Docu...	1 KB
 inline.11613c03bd6d2e4cd6d6.bundle.js	7/20/2017 10:32 AM	JavaScript File	2 KB
 main.8f49e9573568b29c8697.bundle.js	7/20/2017 10:32 AM	JavaScript File	7 KB
 polyfills.c7f2b30f4d8ff16ddd31.bundle.js	7/20/2017 10:32 AM	JavaScript File	95 KB
 styles.6dd9273f3654f3483216.bundle.css	7/20/2017 10:32 AM	Cascading Style Shee...	107 KB
 vendor.db8b86c0f30bd419d13d.bundle.js	7/20/2017 10:32 AM	JavaScript File	299 KB

Observe the new addition of the scripts file, weighing in at 139KB.
(after adding the scripts to the ".angular-cli.json" file.)

 3rdpartylicenses.txt	7/20/2017 11:15 AM	Text Document	0 KB
 favicon.ico	7/20/2017 11:15 AM	Icon	6 KB
 index.html	7/20/2017 11:15 AM	Chrome HTML Docu...	1 KB
 inline.aeae4e0b7160b7241c9.bundle.js	7/20/2017 11:15 AM	JavaScript File	2 KB
 main.b6117d5401b7e27f1cf7.bundle.js	7/20/2017 11:15 AM	JavaScript File	7 KB
 polyfills.c7f2b30f4d8ff16ddd31.bundle.js	7/20/2017 11:15 AM	JavaScript File	95 KB
 scripts.3ec5fc88cd00cf4c8685.bundle.js	7/20/2017 11:15 AM	JavaScript File	139 KB
 styles.cae5a466adf35f725817.bundle.css	7/20/2017 11:15 AM	Cascading Style Shee...	132 KB
 vendor.a8dd41f9b5db597bf5ce.bundle.js	7/20/2017 11:15 AM	JavaScript File	299 KB

See the following angular-cli documentation for more information:

<https://github.com/angular/angular-cli/wiki/stories-third-party-lib>

Swap Your Gear When It's Hot

Thursday, July 20, 2017 11:28 AM

We can make the performance of the developer experience even faster by implementing "hot module replacement".

In the **src/environments** folder, create a new file called:

environment.hmr.ts

Add the following code to this file:

```
1  export const environment = {  
2      production: false,  
3      hmr: true  
4  };  
5  |
```

```
export const environment = {  
  production: false,  
  hmr: true  
};
```

In the **src/environments** folder, edit the **environment.prod.ts** file:

```
1  export const environment = {  
2 +  production: true,  
3 +  hmr: false  
4  };  
5  |
```

Add the following new line:

```
hmr: false
```

In the **src/environments** folder, edit the **environment.ts** file:

```
6  export const environment = {  
7 +  production: false,  
8 +  hmr: false  
9  };  
10 |
```

Add the following new line:

```
hmr: false
```

Edit the **.angular-cli.json**, and add the following:

```
29     "environmentSource": "environments/environment.ts",
30     "environments": {
31       "dev": "environments/environment.ts",
32 +     "hmr": "environments/environment.hmr.ts",
33       "prod": "environments/environment.prod.ts"
34     }

```

"hmr": "environments/environment.hmr.ts",

Edit the package.json file, and add the following:

```
5   "scripts": {
6     "ngver": "ng -v",
7     "start": "ng serve --open",
8     "start:prod": "ng serve --prod --open",
9 +   "hmr": "ng serve --hmr -e=hmr --open",
10    "build": "ng build",
11    "build:prod": "ng build --prod",
12    "build:prod:folder": "ng build --prod --base-href=/my-app/",
13    "test": "ng test",
14    "lint": "ng lint --type-check --format stylish",
15    "lint:fix": "ng lint --type-check --fix --format stylish",
16    "lint:ci": "ng lint --type-check",
17    "e2e": "ng e2e"
18  },

```

"hmr": "ng serve --hmr -e=hmr --open",

Install the angular hmr module dependency.

npm install --save-dev @angularclass/hmr

Or

yarn add @angularclass/hmr --dev

In the src folder, create a new file called:

hmr.ts

Add the following code to this file:

```
import { NgModuleRef, ApplicationRef } from '@angular/core';
import { createNewHosts } from '@angularclass/hmr';

export const hmrBootstrap = (module: any, bootstrap: () => Promise<NgModuleRef<any>>) => {
  let ngModule: NgModuleRef<any>;
  module.hot.accept();
  bootstrap().then(mod => ngModule = mod);
  module.hot.dispose(() => {
    const appRef: ApplicationRef = ngModule.injector.get(ApplicationRef);
    const elements = appRef.components.map(c => c.location.nativeElement);
    const makeVisible = createNewHosts(elements);
    ngModule.destroy();
    makeVisible();
  })
}
```

```
});  
};
```

In the src folder, edit the main.ts file and replace its contents with the following:

```
import { enableProdMode } from '@angular/core';  
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';  
  
import { AppModule } from './app/app.module';  
import { environment } from './environments/environment';  
  
import { hmrBootstrap } from './hmr';  
  
if (environment.production) {  
  enableProdMode();  
}  
  
const bootstrap = () => platformBrowserDynamic().bootstrapModule(AppModule);  
  
if (environment.hmr) {  
  if (module[ 'hot' ]) {  
    hmrBootstrap(module, bootstrap);  
  }  
  else {  
    console.error('HMR is not enabled for webpack-dev-server!');  
    console.log('Are you using the --hmr flag for ng serve?');  
  }  
}  
else {  
  bootstrap();  
}
```

Run the lint:fix script

This time I received and error because the default rule say I should not have the else statements on the next line following the previous block closing brace.

```
C:\srcCC\camp-comfy>npm run lint:fix  
> camp-comfy@0.0.0 lint:fix C:\srcCC\camp-comfy  
> ng lint --type-check --fix --format stylish  
  
C:/srcCC/camp-comfy/src/main.ts  
ERROR: 24:1 one-line  misplaced 'else'  
ERROR: 19:3 one-line  misplaced 'else'  
  
Lint errors found in the listed files.  
npm ERR! code ELIFECYCLE  
npm ERR! errno 2  
npm ERR! camp-comfy@0.0.0 lint:fix: `ng lint --type-check --fix --format stylish`  
npm ERR! Exit status 2  
npm ERR!  
npm ERR! Failed at the camp-comfy@0.0.0 lint:fix script.  
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.  
  
npm ERR! A complete log of this run can be found in:  
npm ERR!     C:\Users\dmoon\AppData\Roaming\npm-cache\_logs\2017-07-20T17_23_58_144Z-debug.log
```

I personally do not agree with this rule, so I change it.

Edit the tslint.json file.

```
75     "one-line": [
76 +     false,
77     "check-open-brace",
78     "check-catch",
79     "check-else",
80     "check-whitespace"
81   ],
```

Change the "one-line" setting from true to false.

Run the lint:fix script again.

```
C:\srcCC\camp-comfy>npm run lint:fix
> camp-comfy@0.0.0 lint:fix C:\srcCC\camp-comfy
> ng lint --type-check --fix --format stylish

All files pass linting.
```

No the code passes all linting rule we have established.

Edit the app.component.html file, as follows:

Change this

```
21 -<button class="btn btn-primary">Test Button</button>
```

To this

```
21 +<div class="form-inline col-md-3">
22 +  <div class="form-group">
23 +    <input type="text" class="form-control"> <button class="btn btn-primary">Test Button</button>
24 +
25 +</div>
```

```
<div class="form-inline col-md-3">
  <div class="form-group">
    <input type="text" class="form-control"> <button class="btn btn-primary">Test Button</button>
</div>
```

Test The Form Using Normal Server Mode

Run the start script.

```
npm run start
```

Enter a value into the input field.

Edit the primary color in the _variables.scss file.

See how the page refreshes and the value entered is cleared.

Test The Form Using Hot Module Replacement Mode

Start the hrm script.

```
npm run hrm
```

Enter a value into the input field.

Edit the primary color in the _variables.scss file.

Observe how it is a little bit faster than the standard server.

See how it does not clear out the input field value.

See the following angular-cli documentation for more information:

<https://github.com/angular/angular-cli/wiki/stories-configure-hmr>

Test Your Gear...Receive Good Karma

Friday, July 21, 2017 4:17 PM

Run the test script.

```
npm run test
```

```
> camp-comfy@0.0.0 test C:\srcCC\camp-comfy
> ng test

10% building modules 1/1 modules 0 active 20 07 2017 09:04:13.904:WARN [karma]: No captured browser, open http://localhost:9876/
20 07 2017 09:04:13.919:INFO [karma]: Karma v1.7.0 server started at http://0.0.0.0:9876/
20 07 2017 09:04:13.928:INFO [launcher]: Launching browser Chrome with unlimited concurrency
20 07 2017 09:04:13.927:INFO [launcher]: Starting browser Chrome 20 07 2017 09:04:23.345:WARN [karma]: No captured browser, open http://localhost:9876/
20 07 2017 09:04:23.866:INFO [Chrome 59.0.3071 (Windows 10 0.0.0)]: Connected on socket 64pCcz84GRw_qXtQAAAA with id 62278244
Chrome 59.0.3071 (Windows 10 0.0.0) AppComponent should have as title 'app' FAILED
  Expected 'Camp Comfy' to equal 'app'.
    at Object.<anonymous> (http://localhost:9876/_karma_webpack_/main.bundle.js:93:27)
    at ZoneDelegate.webpackJsonp.../.../.../zone.js/dist/zone.js.ZoneDelegate.invoke (http://localhost:9876/_karma_webpack_/polyfills.bundle.js:2801:26)
    at AsyncTestZoneSpec.webpackJsonp.../.../.../zone.js/dist/async-test.js.AsyncTestZoneSpec.onInvoke (http://localhost:9876/_karma_webpack_/vendor.bundle.js:4656:39)
```

This time we get errors.

Karma v1.7.0 - connected

DEBUG

Chrome 59.0.3071 (Windows 10 0.0.0) is idle

Jasmine 2.6.4

finished in 0.289s

• ✘ ✘

3 specs, 2 failures

raise exceptions

[Spec List](#) | [Failures](#)

AppComponent should have as title 'app'

Expected 'Camp Comfy' to equal 'app'.

Error: Expected 'Camp Comfy' to equal 'app'.

```
at stack (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at buildExpectationResult (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at Spec.expectationResultFactory (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at Spec.addExpectationResult (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at Expectation.addExpectationResult (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at Expectation.toEqual (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at Object.<anonymous> (http://localhost:9876/_karma_webpack_/main.bundle.js:93:27)
at ZoneDelegate.webpackJsonp.../.../.../zone.js/dist/zone.js.ZoneDelegate.invoke (http://localhost:9876/_karma_webpack_/polyfills.bundle.js:2801:26)
at AsyncTestZoneSpec.webpackJsonp.../.../.../zone.js/dist/async-test.js.AsyncTestZoneSpec.onInvoke (http://localhost:9876/_karma_webpack_/vendor.bundle.js:4656:39)
at ProxyZoneSpec.webpackJsonp.../.../.../zone.js/dist/proxy.js.ProxyZoneSpec.onInvoke (http://localhost:9876/_karma_webpack_/vendor.bundle.js:500:27)
```

AppComponent should render title in a h1 tag

Expected '

Welcome to Camp Comfy!

' to contain 'Welcome to app!'.

Error: Expected '

Welcome to Camp Comfy!

' to contain 'Welcome to app!'.

```
at stack (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at buildExpectationResult (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at Spec.expectationResultFactory (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
at Spec.addExpectationResult (http://localhost:9876/base/node_modules/jasmine-core/lib/jasmine-core/jasmine.js?da99c5b057693d025fad3d7685e1590600ca376d:2176:17)
```

It reports that our application title does not match the expected value 'app'.

Since we want our new application title, we must modify our test scripts to accommodate the changes.

Edit the **app.component.spec.ts** file.

```
1 import { TestBed, async } from '@angular/core/testing';
2 import { RouterTestingModule } from '@angular/router/testing';
3
4 import { AppComponent } from './app.component';
5
6 describe('AppComponent', () => {
7   beforeEach(async(() => {
8     TestBed.configureTestingModule({
9       imports: [
10         RouterTestingModule
11       ],
12       declarations: [
13         AppComponent
14       ],
15     }).compileComponents();
16   }));
17
18   it('should create the app', async(() => {
19     const fixture = TestBed.createComponent(AppComponent);
20     const app = fixture.debugElement.componentInstance;
21     expect(app).toBeTruthy();
22   }));
23
24   it(`should have as title 'app'`, async(() => {
25     const fixture = TestBed.createComponent(AppComponent);
26     const app = fixture.debugElement.componentInstance;
27     expect(app.title).toEqual('app');
28   }));
29
30   it('should render title in a h1 tag', async(() => {
31     const fixture = TestBed.createComponent(AppComponent);
32     fixture.detectChanges();
33     const compiled = fixture.debugElement.nativeElement;
34     expect(compiled.querySelector('h1').textContent).toContain('Welcome to app!');
35   }));
36 });
37
```

Change of the title text 'app' to "Camp Comfy"

```

4 import { AppComponent } from './app.component';
5
6 describe('AppComponent', () => {
7   beforeEach(async(() => {
8     TestBed.configureTestingModule({
9       imports: [
10         RouterTestingModule
11       ],
12       declarations: [
13         AppComponent
14       ],
15     }).compileComponents();
16   }));
17
18   it('should create the app', async(() => {
19     const fixture = TestBed.createComponent(AppComponent);
20     const app = fixture.debugElement.componentInstance;
21     expect(app).toBeTruthy();
22   }));
23
24 + it(`should have as title 'Camp Comfy'`, async(() => {
25   const fixture = TestBed.createComponent(AppComponent);
26   const app = fixture.debugElement.componentInstance;
27 +   expect(app.title).toEqual('Camp Comfy');
28   }));
29
30   it('should render title in a h1 tag', async(() => {
31     const fixture = TestBed.createComponent(AppComponent);
32     fixture.detectChanges();
33     const compiled = fixture.debugElement.nativeElement;
34 +   expect(compiled.querySelector('h1').textContent).toContain('Welcome to Camp Comfy!');
35   }));
36 });
37

```

Rerun the test script.

npm run test

```
C:\srcCC\camp-comfy>npm run test
> camp-comfy@0.0.0 test C:\srcCC\camp-comfy
> ng test

10% building modules 1/1 modules 0 active
20 07 2017 09:09:45.565:WARN [karma]: No captured browser, open http://localhost:9876/
20 07 2017 09:09:45.577:INFO [karma]: Karma v1.7.0 server started at http://0.0.0.0:9876/
20 07 2017 09:09:45.578:INFO [launcher]: Launching browser Chrome with unlimited concurrency
20 07 2017 09:09:45.583:INFO [launcher]: Starting browser Chrome
20 07 2017 09:09:54.628:WARN [karma]: No captured browser, open http://localhost:9876/
20 07 2017 09:09:55.112:INFO [Chrome 59.0.3071 (Windows 10 0.0.0)]: Connected on socket xnEPYm8YCzpneRvAAAAAA with id 87715894
Chrome 59.0.3071 (Windows 10 0.0.0): Executed 3 of 3 SUCCESS (0.143 secs / 0.263 secs)
```

The tests should now all pass.

Karma v1.7.0 - connected

DEBUG

Chrome 59.0.3071 (Windows 10 0.0.0) is idle

Jasmine 2.6.4

finished in 0.27s

• • •

3 specs, 0 failures

raise exceptions

```
AppComponent
  should create the app
  should have as title 'Camp Comfy'
  should render title in a h1 tag
```

Welcome to Camp Comfy!



Enhance Your Karma Capabilities

Friday, July 21, 2017 5:15 PM

Add Chrome Headless Support

Modify your Karma configuration to support running chrome in the headless mode to support Continuous Integration (CI) testing.

Edit the **karma.conf.js** file.

Add the following section after the plugins section:

```
customLaunchers: {  
  ChromeNoSandboxHeadless: {  
    base: 'Chrome',  
    flags: [  
      '--no-sandbox',  
      // See https://chromium.googlesource.com/chromium/src/+/lkgr/headless/README.md  
      '--headless',  
      '--disable-gpu',  
      // Without a remote debugging port, Google Chrome exits immediately.  
      '--remote-debugging-port=9222',  
    ],  
  },  
},
```

Add Code Coverage Configuration Information

Modify your Karma configuration to support outputting Code Coverage reports to support Continuous Integration testing.

Edit the **karma.conf.js** file.

In the plugins section, add the following statement:

```
require('karma-junit-reporter'),
```

Add the following after the client section:

In the coverageIstanbulReporter section, modify the reports value, as follows:

Modify the reports section to look as follows:

```
reporters: config.angularCli && config.angularCli.codeCoverage  
? ['progress', 'coverage-istanbul']  
: ['progress', 'kjhtml'],
```

Example Updated karma.conf.js File

The Karma configuration file should now look like the following:

```
// Karma configuration file, see link for more information  
// https://karma-runner.github.io/0.13/config/configuration-file.html  
  
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('karma-junit-reporter'),
    require('@angular/cli/plugins/karma')
  ],
  customLaunchers: {
    ChromeNoSandboxHeadless: {
      base: 'Chrome',
      flags: [
        '--no-sandbox',
        // See https://chromium.googlesource.com/chromium/src/+/lkgr/headless/README.md
        '--headless',
        '--disable-gpu',
        // Without a remote debugging port, Google Chrome exits immediately.
        '--remote-debugging-port=9222',
      ],
    },
  },
  client: {
    clearContext: false // leave Jasmine Spec Runner output visible in browser
  },
  junitReporter: {
    outputDir: 'junitResults',
    suite: 'models'
  },
  coverageIstanbulReporter: {
    reports: ['html', 'lcovonly', 'cobertura'],
    fixWebpackSourcePaths: true
  },
  angularCli: {
    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
});
});
};

```

Install Code Coverage Reporting Package

Install the Junit code coverage reporting tool.

Open a command line window and run the following:

```
npm install karma-junit-reporter --save-dev
```

Or

```
yarn add karma-junit-reporter --dev
```

You are now configured to support the generation of code coverage reports for your CI builds.

Make Sure Your Stuff Is Covered

Friday, July 21, 2017 5:06 PM

Add Test Script With Code Coverage For CI Builds

Edit the **package.json** file and add the following to the scripts section:

```
"test:ci": "ng test --browsers=ChromeNoSandboxHeadless --watch=false --single-run=true --code-coverage=true --reporters progress,junit,coverage-istanbul",
```

Run Code Coverage Reports

Run the test:ci script from the command line.

You should see output like the following:

```
C:\srcCC\camp-comfy>npm run test:ci

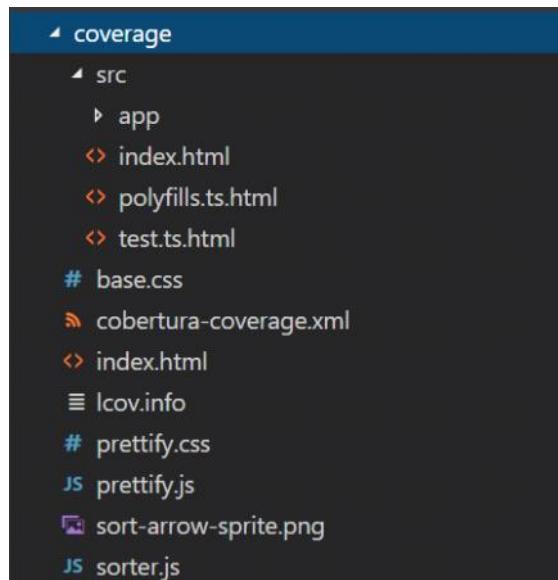
> camp-comfy@0.0.0 test:ci C:\srcCC\camp-comfy
> ng test --browsers=ChromeNoSandboxHeadless --watch=false --single-run=true --code-coverage=true --reporters progress,junit,coverage-istanbul

  10% building modules 2/2 modules 0 active21 07 2017 16:40:17.444:INFO [karma]: Karma v1.7.0 server started at http://0.0.0.0:9876/
21 07 2017 16:40:17.447:INFO [launcher]: Launching browser ChromeNoSandboxHeadless with unlimited concurrency
21 07 2017 16:40:17.453:INFO [launcher]: Starting browser Chrome 21 07 2017 16:40:28.589:INFO [Chrome 59.0.3071 (Windows 10 0.0.0)]: Connected on socket o7bcIecpoiVEG3UbAAAA with id 12624116
Chrome 59.0.3071 (Windows 10 0.0.0): Executed 3 of 3 SUCCESS (0.244 secs / 0.235 secs)

C:\srcCC\camp-comfy>
```

In the root of your angular project, you should see a new folder called coverage.

It contains the code coverage report for the unit tests that ran.



Launch the index.html file in a browser, and you should see something like the following:

All files

100% Statements 36/36 100% Branches 0/0 100% Functions 3/3 100% Lines 34/34

File	Statements	Branches	Functions	Lines
src	100%	30/30	100%	1/1
src/app	100%	6/6	100%	2/2

You can drill into the code sections for more details.

All files src/app

100% Statements 6/6 100% Branches 0/0 100% Functions 2/2 100% Lines 4/4

File	Statements	Branches	Functions	Lines
app.component.ts	100%	6/6	100%	2/2

All files / src/app app.component.ts

100% Statements 6/6 100% Branches 0/0 100% Functions 2/2 100% Lines 4/4

```
1 1x import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
8 1x export class AppComponent {
9 3x   title = 'Camp Comfy';
10 1x }
```

Your Camp Mascot Is A Wallaby

Friday, July 21, 2017 2:46 PM

Wallaby.js runs your JavaScript tests immediately as you type and displays execution results in your code editor. It also provides live test execution and code coverage reports.

This tool increases developer productivity when writing JavaScript unit and integration tests.

This adds business value by saving the business money due to the reduced time required to develop the tests.

You can get a copy from the Wallaby.js web site:

<https://wallabyjs.com/>

Visual Studio Code Installation Procedure

<https://wallabyjs.com/docs/intro/install.html#visual-studio-code>

Add Wallaby.js support to your angular-cli project

wallaby.js

In the root of your Angular project, create a new file called:

wallaby.js

Copy the following code to this file:

```
var wallabyWebpack = require('wallaby-webpack');
var path = require('path');

var compilerOptions = Object.assign(
  require('./tsconfig.json').compilerOptions,
  require('./src/tsconfig.spec.json').compilerOptions);

module.exports = function (wallaby) {

  var webpackPostprocessor = wallabyWebpack({
    entryPatterns: [
      'src/wallabyTest.js',
      'src/**/*spec.js'
    ],
    module: {
      loaders: [
        {test: /\.css$/, loader: ['raw-loader', 'css-loader']},
        {test: /\.html$/, loader: 'raw-loader'},
        {test: /\.ts$/, loader: '@ngtools/webpack', include: /node_modules/, query: {tsConfigPath: 'tsconfig.json'}},
        {test: /\.js$/, loader: 'angular2-template-loader', exclude: /node_modules/},
        {test: /\.json$/, loader: 'json-loader'},
        {test: /\.styl$/, loaders: ['raw-loader', 'stylus-loader']},
        {test: /\.less$/, loaders: ['raw-loader', 'less-loader']},
        {test: /\.scss$|\.sass$/, loaders: ['raw-loader', 'sass-loader']},
        {test: /\.jpg|png$/, loader: 'url-loader?limit=128000'}
      ]
    },
    resolve: {
      extensions: ['.js', '.ts'],
      modules: [
        path.join(wallaby.projectCacheDir, 'src/app'),
        path.join(wallaby.projectCacheDir, 'src'),
        'node_modules'
      ]
    },
    node: {
      fs: 'empty',
    }
  });
}
```

```

        net: 'empty',
        tls: 'empty',
        dns: 'empty'
    });
}

return {
    files: [
        {pattern: 'src/**/*.ts', load: false},
        {pattern: 'src/**/*.d.ts', ignore: true},
        {pattern: 'src/**/*.css', load: false},
        {pattern: 'src/**/*.less', load: false},
        {pattern: 'src/**/*.scss', load: false},
        {pattern: 'src/**/*.sass', load: false},
        {pattern: 'src/**/*.styl', load: false},
        {pattern: 'src/**/*.html', load: false},
        {pattern: 'src/**/*.json', load: false},
        {pattern: 'src/**/*spec.ts', ignore: true}
    ],
    tests: [
        {pattern: 'src/**/*spec.ts', load: false}
    ],
    testFramework: 'jasmine',
    compilers: {
        '**/*.ts': wallaby.compilers.typeScript(compilerOptions)
    },
    middleware: function (app, express) {
        var path = require('path');
        app.use('/favicon.ico', express.static(path.join(__dirname, 'src/favicon.ico')));
        app.use('/assets', express.static(path.join(__dirname, 'src/assets')));
    },
    env: {
        kind: 'chrome'
    },
    postprocessor: webpackPostprocessor,
    setup: function () {
        window.__moduleBundler.loadTests();
    },
    debug: true
};
};

```

wallabyTest.ts

In the src folder, create the following new file:

wallabyTest.ts

Add the following code to this file:

```

import './polyfills';
import 'zone.js/dist/long-stack-trace-zone';
import 'zone.js/dist/proxy.js';
import 'zone.js/dist/sync-test';
import 'zone.js/dist/jasmine-patch';
import 'zone.js/dist/async-test';
import 'zone.js/dist/fake-async-test';
import { getTestBed } from '@angular/core/testing';
import {
    BrowserDynamicTestingModule,
    platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';
getTestBed().initTestEnvironment(
    BrowserDynamicTestingModule,

```

```
platformBrowserDynamicTesting()  
);
```

In the src folder, edit the **tsconfig.app.ts** file.

```
1  {  
2    "extends": "../tsconfig.json",  
3    "compilerOptions": {  
4      "outDir": "../out-tsc/app",  
5      "baseUrl": "./",  
6      "module": "es2015",  
7      "types": []  
8    },  
9    "exclude": [  
10      "test.ts",  
11      "wallabyTest.ts",  
12      "**/*spec.ts"  
13    ]  
14  }  
15
```

Add the following entry to the exclude section:

```
src/wallabyTest.ts
```

Open a command line window and run the following:

```
npm install wallaby-webpack angular2-template-loader electron --save-dev
```

```
C:\srcCC\camp-comfy>npm install wallaby-webpack angular2-template-loader electron --save-dev  
> electron@1.6.11 postinstall C:\srcCC\camp-comfy\node_modules\electron  
> node install.js  
  
npm [WARN optional] SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules\fsevents):  
npm [WARN notsup] SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})  
  
+ wallaby-webpack@0.0.38  
+ electron@1.6.11  
+ angular2-template-loader@0.6.2  
added 156 packages in 39.909s
```

Install Typescript and TSLint

Both Typescript and TSLint are required to be globally installed for it to successfully function.

Install typescript globally.

```
npm install -g typescript@latest  
Or  
yarn global add typescript@latest
```

Install tslint globally.

```
npm install -g tslint@latest  
Or  
yarn global add tslint@latest
```

The **package.json** file devDependencies sectionshould have two new entries, as follows:

Install additional development dependencies for wallaby.js

```
npm install -g wallaby-webpack  
Or
```

```
yarn global add wallaby-webpack
```

```
npm install -g angular2-template-loader
```

Or

```
yarn global add angular2-template-loader
```

For more information, see the following:

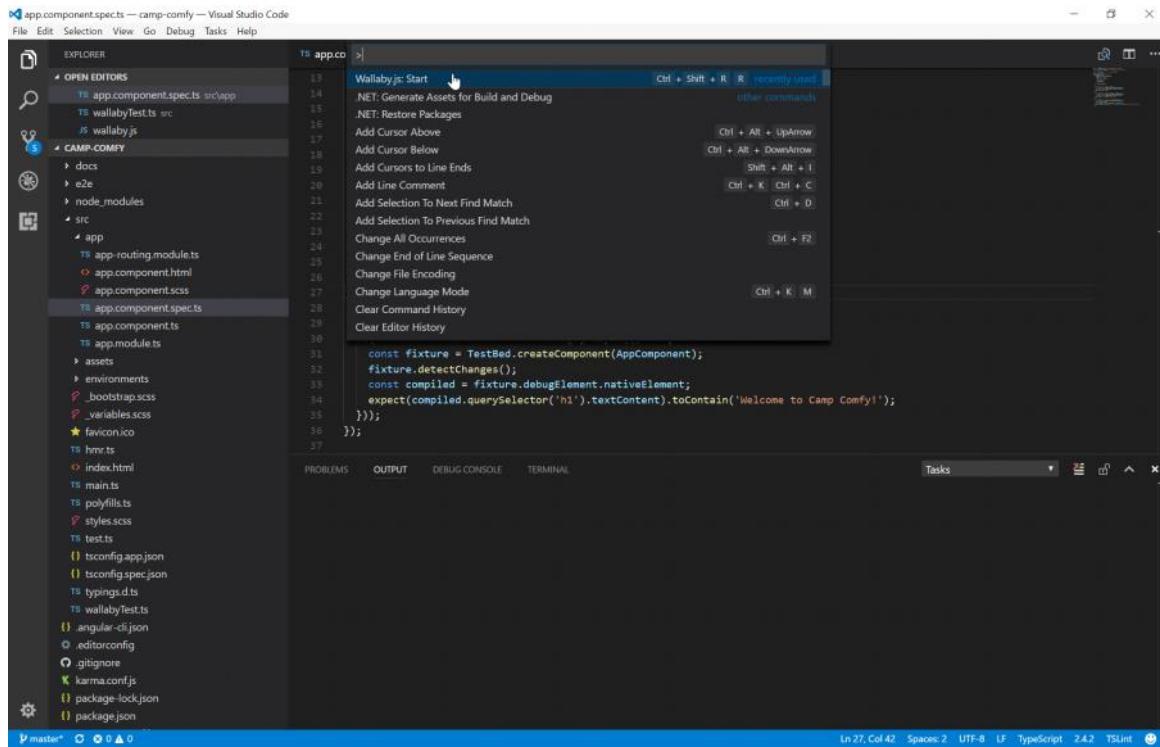
<https://wallabyjs.com/>

<https://github.com/wallabyjs/ngCliWebpackSample>

I recommend that you do not install electro as this adds about 50MB of additional files that you do not need.

You should use the Chrome headless option when running wallaby.js.

To run Wallaby.js in Visual Studio Code, set focus to VS Code and type CTRL+SHIFT+P to show the Command Palette.



Type **Wallaby.js** and select the **Wallaby.js Start** command.

In the lower right corner of the VS Code window, you should see the busy indicator for Wallaby.js working



When the unit tests have completed, you should see something like the following if the tests all pass:

```

    13     AppComponent
    14   );
    15   }).compileComponents();
    16 });
    17 });
    18 it('should create the app', async(() => {
    19   const fixture = TestBed.createComponent(AppComponent);
    20   const app = fixture.debugElement.componentInstance;
    21   expect(app).toBeTruthy();
    22 });
    23 });
    24 it('should have as title 'Camp Comfy', async(() => {
    25   const fixture = TestBed.createComponent(AppComponent);
    26   const app = fixture.debugElement.componentInstance;
    27   expect(app.title).toEqual('Camp Comfy');
    28 });
    29 });
    30 it('should render title in a h1 tag', async(() => {
    31   const fixture = TestBed.createComponent(AppComponent);
    32   fixture.detectChanges();
    33   const compiled = fixture.debugElement.nativeElement;
    34   expect(compiled.querySelector('h1').textContent).toContain('Welcome to Camp Comfy');
    35 });
    36 });
    37 });

```

The green dots in the code window indicate the test ran the code and it passed the test.

The output window below indicate information regarding the status of the tests.

In the lower right corner of the VS Code window, you should see the Wallaby.js test run status indicator shows 0 tests failed, and 3 tests passed.

TypeScript 2.4.2 X 0 ✓ 3

Now change one of the tests to a failing condition.

Wallaby.js automatically runs the test for the changed application or test code.

```

    24 it('should have as title 'Camp Comfy', async(() => {
    25   const fixture = TestBed.createComponent(AppComponent);
    26   const app = fixture.debugElement.componentInstance;
    27   expect(app.title).toEqual('Camp Comfy Bad'); Expected 'Camp Comfy' to equal 'Camp Comfy Bad'.
    28 });
    29 });

```

The tests indicate the failed assertion.

In the lower right corner of the VS Code window, you should see the Wallaby.js test run status indicator shows 1 test failed, and only 2 tests passed, and shows an error message

TypeScript 2.4.2 X 1 ✓ 2 Expected 'Camp Comfy' to equal 'Camp Comfy Bad'.

Navigate to an application code file, and the status will change, as follows

TypeScript 2.4.2 X 1 ✓ 2

In the lower right corner of the VS Code window, you should see the Wallaby.js test run status indicator.

Click on it and the Wallaby.js Tests console window appears showing information regarding the failing test(s).

A screenshot of a code editor interface, specifically showing the results of a Wallaby.js test run. The top navigation bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and a dropdown menu for Wallaby.js Tests. The main content area displays the following test results:

```
| failing tests, 2 passing

AppComponent should have as title 'Camp Comfy' [126 ms]

Camp Comfy_Bad

Expected 'Camp Comfy' to equal 'Camp Comfy Bad'.
at Object.<anonymous> src/app/app.component.spec.ts:27:4
```

Smoke Test Your Camp Site End-to-End

Friday, July 21, 2017 5:19 PM

Run the end-to-end test script (e2e)

```
*****
*          Failures          *
*****  
1) camp-comfy App should display welcome message
  - Expected 'Welcome to Camp Comfy!' to equal 'Welcome to app!'.  
  
Executed 1 of 1 spec (1 FAILED) in 1 sec.  
[09:15:27] I/launcher - 0 instance(s) of WebDriver still running  
[09:15:27] I/launcher - chrome #01 failed 1 test(s)  
[09:15:27] I/launcher - overall: 1 failed spec(s)  
[09:15:27] E/launcher - Process exited with error code 1  
npm ERR! code ELIFECYCLE  
npm ERR! errno 1  
npm ERR! camp-comfy@0.0.0 e2e: `ng e2e`  
npm ERR! Exit status 1  
npm ERR!  
npm ERR! Failed at the camp-comfy@0.0.0 e2e script.  
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.  
  
npm ERR! A complete log of this run can be found in:  
npm ERR!     C:\Users\dmoon\AppData\Roaming\npm-cache\_logs\2017-07-20T14_15_27_313Z-debug.log
```

The test will report an error.

Edit the **app.e2e-spec.ts** file.

```
1  import { CampComfyPage } from './app.po';  
2  
3  describe('camp-comfy App', () => {  
4    let page: CampComfyPage;  
5  
6    beforeEach(() => {  
7      page = new CampComfyPage();  
8    });  
9  
10   it('should display welcome message', () => {  
11     page.navigateTo();  
12     expect(page.getParagraphText()).toEqual('Welcome to app!');  
13   });  
14});  
15
```

Change the application title test to indicate "Camp Comfy".

```

1 import { CampComfyPage } from './app.po';
2
3 describe('camp-comfy App', () => {
4   let page: CampComfyPage;
5
6   beforeEach(() => {
7     page = new CampComfyPage();
8   });
9
10  it('should display welcome message', () => {
11    page.navigateTo();
12 +   expect(page.getParagraphText()).toEqual('Welcome to Camp Comfy!');
13  });
14});
15

```

Run the e2e test again, and it should now pass.

```

> ng e2e

** NG Live Development Server is listening on localhost:49152, open your browser on http://localhost:49152 **
(node:18124) [DEP0022] DeprecationWarning: os.tmpDir() is deprecated. Use os.tmpdir() instead.
Hash: e79e0157c82f6bc14214
Time: 10460ms
chunk  {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 177 kB {4} [initial] [rendered]
chunk  {1} main.bundle.js, main.bundle.js.map (main) 6.58 kB {3} [initial] [rendered]
chunk  {2} styles.bundle.js, styles.bundle.js.map (styles) 10.7 kB {4} [initial] [rendered]
chunk  {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.48 MB [initial] [rendered]
chunk  {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]
webpack: Compiled successfully.
[09:20:30] I/update - chromedriver: file exists C:\srcCC\camp-comfy\node_modules\protractor\node_modules\webdriver-manager\selenium\chromedriver_2.30.zip
[09:20:30] I/update - chromedriver: unzipping chromedriver_2.30.zip
[09:20:31] I/update - chromedriver: chromedriver_2.30.exe up to date
[09:20:31] I/launcher - Running 1 instances of WebDriver
[09:20:31] I/direct - Using ChromeDriver directly...
Jasmine started

  camp-comfy App
    ✓ should display welcome message

Executed 1 of 1 spec SUCCESS in 1 sec.
[09:20:35] I/launcher - 0 instance(s) of WebDriver still running
[09:20:35] I/launcher - chrome #01 passed

```

Enable Protractor To Run Headless

Add a new file to the root of your angular project called:

protractor-ci.conf.js

Add the following code to this file.

```

// Protractor configuration file, see link for more information
// https://github.com/angular/protractor/blob/master/lib/config.ts

const { SpecReporter } = require('jasmine-spec-reporter');

exports.config = {
  allScriptsTimeout: 11000,
  specs: [

```

```
'./e2e/**/*.e2e-spec.ts'  
],  
capabilities: {  
  browserName: 'chrome',  
  chromeOptions: {  
    args: ["--headless", "--disable-gpu", "--window-size=800x600"]  
  }  
},  
directConnect: true,  
baseUrl: 'http://localhost:4200/',  
framework: 'jasmine',  
jasmineNodeOpts: {  
  showColors: true,  
  defaultTimeoutInterval: 30000,  
  print: function() {}  
},  
beforeLaunch: function() {  
  require('ts-node').register({  
    project: 'e2e/tsconfig.e2e.json'  
  });  
},  
onPrepare() {  
  jasmine.getEnv().addReporter(new SpecReporter({ spec: { displayStacktrace: true } }));  
}  
};
```

Edit the package.json file, and add the following to the scripts section.

```
"e2e:ci": "ng e2e --config=protractor-ci.conf.js"
```

Open a command line window and run the e2e:ci script.

The e2e test should run in a headless mode (looks like an empty browser), like the following.

```
C:\srcCC\camp-comfy>npm run e2e:ci

> camp-comfy@0.0.0 e2e:ci C:\srcCC\camp-comfy
> ng e2e --config=protractor-ci.conf.js

** NG Live Development Server is listening on localhost:49152, open your browser on http://localhost:49152 **
(node:38864) [DEP0022] DeprecationWarning: os.tmpDir() is deprecated. Use os.tmpdir() instead.
Hash: da0a5efbbdd797de22b9
Time: 12984ms
chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 252 kB {5} [initial] [rendered]
chunk {1} main.bundle.js, main.bundle.js.map (main) 7.77 kB {4} [initial] [rendered]
chunk {2} scripts.bundle.js, scripts.bundle.js.map (scripts) 146 kB {5} [initial] [rendered]
chunk {3} styles.bundle.js, styles.bundle.js.map (styles) 193 kB {5} [initial] [rendered]
chunk {4} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.49 MB [initial] [rendered]
chunk {5} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]
webpack: Compiled successfully.

[17:35:37] I/update - chromedriver: file exists C:\srcCC\camp-comfy\node_modules\protractor\node_modules\webdriver-manager\selenium\chromedriver_2.30.zip
[17:35:37] I/update - chromedriver: unzipping chromedriver_2.30.zip
[17:35:37] I/update - chromedriver: chromedriver_2.30.exe up to date
[17:35:37] I/launcher - Running 1 instances of WebDriver
[17:35:37] I/direct - Using ChromeDriver directly...
Jasmine started

  camp-comfy App
    ✓ should display welcome message

Executed 1 of 1 spec SUCCESS in 1 sec.
[17:35:41] I/launcher - 0 instance(s) of WebDriver still running
[17:35:41] I/launcher - chrome #01 passed
```

Form A New Check-in Checklist Habit

Sunday, August 6, 2017 8:03 AM

Highly recommend getting your entire camp site team to adopt a new habit of doing the following just prior to checking source code into a version control system:

1. Run the lint:fix script to look for and clean up syntax issues.
2. Run the test script to run through all of your unit and component integration tests.
3. Run the e2e test to run through all of your site UI end-to-end tests.
4. Run the build:prod script to ensure that your code can compile in production mode (to catch AoT ahead of time compile issues).

Take action after step to remediate any issues discovered by the step action.

Getting everyone into these habits from day one of your camp site build will save you many, many hours of troubleshooting undiscovered issues.

Share Your Camp Tools And Services

Saturday, July 22, 2017 9:01 AM

Camp site services, components, directives, pipes, etc., can also be modularized and shared among your camp site locations.

For more information on Angular modules, see the following:

<https://angular.io/guide/ngmodule#shared-modules>

Create A Services Module

Most of your services should be global singletons.

Create a module for globally available services:

ng g m services

```
C:\srcCC\camp-comfy>ng g m services
installing module
  create src\app\services\services.module.ts
  WARNING Module is generated but not provided, it must be provided to be used
```

Edit the new **services.module.ts** file.

Modify the code as shown below:

```
1 + import { NgModule, ModuleWithProviders } from '@angular/core';
2   import { CommonModule } from '@angular/common';
3
4   @NgModule({
5     imports: [
6       CommonModule
7     ]
8   })
9 + export class ServicesModule {
10 +   static forRoot(): ModuleWithProviders {
11 +     return {
12 +       ngModule: ServicesModule,
13 +       providers: [
14 +
15 +     ]
16 +   };
17 + }
18 + }
19 +
```

```
import { NgModule, ModuleWithProviders } from '@angular/core';
import { CommonModule } from '@angular/common';
```

```

@NgModule({
  imports: [
    CommonModule
  ]
})
export class ServicesModule {
  static forRoot(): ModuleWithProviders {
    return {
      ngModule: ServicesModule,
      providers: [
        ...
      ];
    }
  }
}

```

Note: The **static forRoot(): ModuleWithProviders** statement ensures that only one instance of any services declared in the providers sections will be created and made available to all application modules and components. However, you must still import the services wherever needed.

Create A Shared Module

You will likely have many components, directives, pipes, etc. that you'll wish to share with the various feature of your camp site.

Create a module for shared components:

ng g m shared

```
C:\srcCC\camp-comfy>ng g m shared
installing module
  create src\app\shared\shared.module.ts
WARNING Module is generated but not provided, it must be provided to be used
```

Edit the new **shared.module.ts** file.

Modify the code as shown below:

```
1 + import { NgModule, ModuleWithProviders } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4  @NgModule({
5    imports: [
6      CommonModule
7    ],
8 +   declarations: [
9 +
10 + ],
11 +   exports: [
12 +
13 + ]
14 })
15 + export class SharedModule {
16 +   static forRoot(): ModuleWithProviders {
17 +     return {
18 +       ngModule: SharedModule,
19 +       providers: [
20 +
21 +     ]
22 +   };
23 + }
24 + }
25
```

```
import { NgModule, ModuleWithProviders } from '@angular/core';
import { CommonModule } from '@angular/common';
```

```
@NgModule({
  imports: [
    CommonModule
  ],
  declarations: [
  ],
  exports: [
  ]
})
export class SharedModule {
  static forRoot(): ModuleWithProviders {
    return {
      ngModule: SharedModule,
      providers: [
      ]
    };
  }
}
```

Note: The **static forRoot(): ModuleWithProviders** statement ensures that only one instance of any

services declared in the providers sections will be created and made available to all application modules and components. However, you must still import the services wherever needed.

Edit the **app.module.ts** file.

Add the **ServiceModule** and **SharedModule** to the imports section and be sure to include the **.forRoot()** syntax as show below:

```
1 import { BrowserModule, Title } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import './rxjs-extensions';
5 +import { ServicesModule } from './services/services.module';
6 +import { SharedModule } from './shared/shared.module';
7
8 import { AppRoutingModule } from './app-routing.module';
9 import { AppComponent } from './app.component';
10 import { HomeComponent } from './home/home.component';
11 import { ErrorComponent } from './error/error.component';
12 import { NotFoundComponent } from './not-found/not-found.component';
13
14 @NgModule({
15   declarations: [
16     AppComponent,
17     HomeComponent,
18     ErrorComponent,
19     NotFoundComponent
20   ],
21   imports: [
22     BrowserModule,
23 +   ServicesModule.forRoot(),
24 +   SharedModule.forRoot(),
25     AppRoutingModule
26   ],
27   providers: [
28     Title
29   ],
30   bootstrap: [AppComponent]
31 })
32 export class AppModule { }
```

Generate a new flicker service under the services folder and add it to the services module.

```
ng g s services\flickr
```

```
C:\srcCC\camp-comfy>ng g s services\flickr
installing service
  create src\app\services\flickr.service.spec.ts
  create src\app\services\flickr.service.ts
WARNING Service is generated but not provided, it must be provided to be used
```

Edit the new **flickr.service.ts** file, and replace its contents with the following:

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

```

import { Http, Response, RequestOptions, Headers, URLSearchParams } from '@angular/http';
import { Observable } from 'rxjs/Observable';

@Injectable()
export class FlickrService {

  constructor(
    private http: Http
  ) { }

  baseUrl: string = 'https://api.flickr.com/services/rest';

  private photos: any[] = [];

  public getPhotos(): any[] {
    return this.photos;
  }

  search(searchText: string): Observable<any> {

    console.log(searchText);

    const params = new URLSearchParams();
    params.set('method', 'flickr.photos.search');
    params.set('format', 'json');
    // Replace the following with your flicker API key
    params.set('api_key', '53286431b01976732160ae74b1c81a8b');
    params.set('action', 'opensearch');
    params.set('text', searchText);
    params.set('per_page', '25');
    params.set('media', 'photos');
    params.set('content_type', '1');
    params.set('format', 'json');
    params.set('nojsoncallback', '1');

    const requestOptions = new RequestOptions();
    requestOptions.search = params;

    return this.http.get(this.baseUrl, requestOptions)
      .map(this.extractData)
      .catch(this.handleError);
  }

  private extractData(res: Response): any {

    this.photos = [];

    const respPhotos = res.json().photos.photo;

    for (let i = 0; i < respPhotos.length; i++) {

      const farm = respPhotos[i].farm;

```

```

const server = respPhotos[i].server;
const id = respPhotos[i].id;
const secret = respPhotos[i].secret;
const title = respPhotos[i].title;

const photoUrl = `https://farm${farm}.staticflickr.com/${server}/${id}_${secret}.jpg`;

let titleText = title;

const pos = title.indexOf(' #');

if (pos > 0) {
  titleText = title.substring(0, pos - 1);
}

this.photos.push({ url: photoUrl, title: titleText });

}

return res.json().photos.photo;
}

private handleError(error: Response | any) {

// In a real world app, you might use a remote logging infrastructure
let errMsg: string;

if (error instanceof Response) {
  const body = error.json() || "";
  const err = body.error || JSON.stringify(body);
  errMsg = `${error.status} - ${error.statusText || ""} ${err}`;
}
else {
  errMsg = error.message ? error.message : error.toString();
}

console.error(errMsg);

return Observable.throw(errMsg);
}
}

```

Edit the **services.module.ts** file.

Add the **FlickrService** to the providers and import the service as show below:

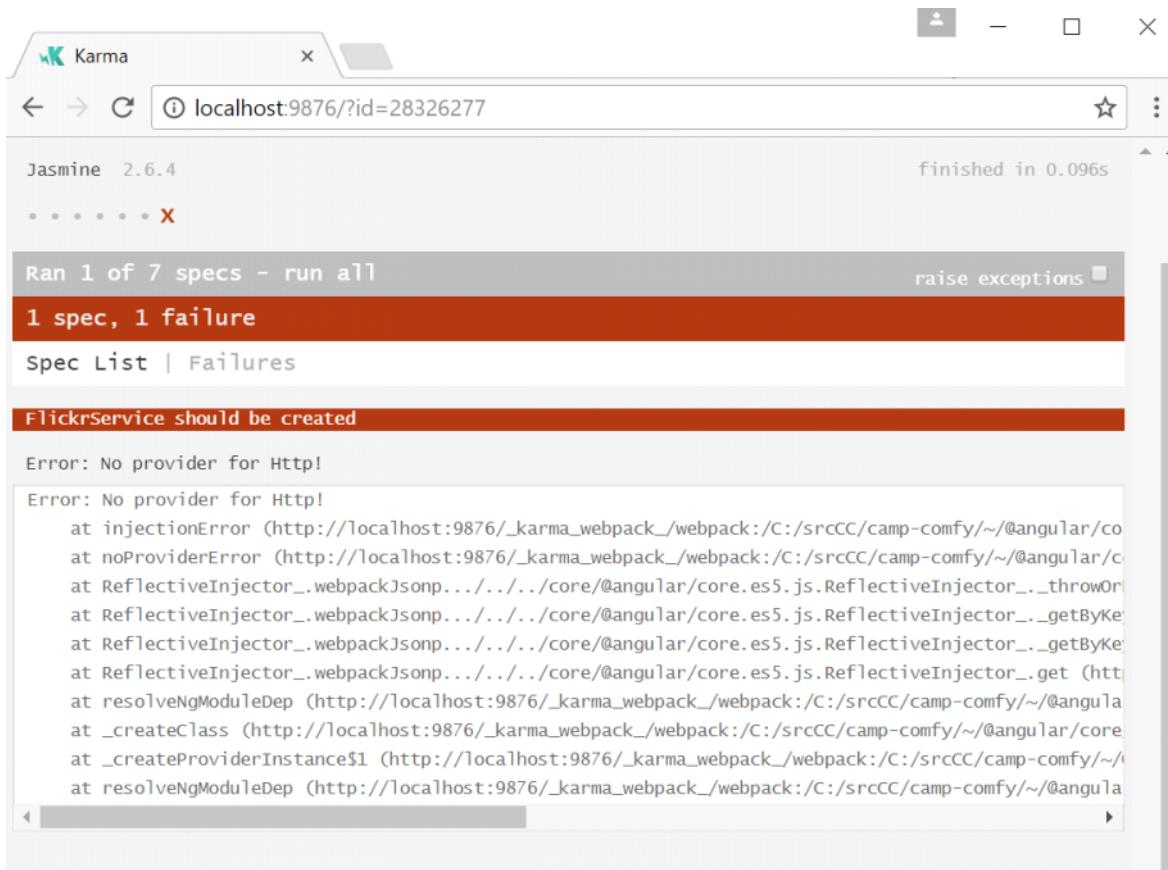
```
1 import { NgModule, ModuleWithProviders } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 +import { FlickrService } from './flickr.service';
5 +
6 @NgModule({
7   imports: [
8     CommonModule
9   ]
10 })
11 export class ServicesModule {
12   static forRoot(): ModuleWithProviders {
13     return {
14       ngModule: ServicesModule,
15       providers: [
16 +       FlickrService
17     ]
18   };
19 }
20 }
21
```

Edit the **services.module.ts** file.

Add a provider entry for the FlickrService and modify the code as shown below:

```
1 import { NgModule, ModuleWithProviders } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 +import { FlickrService } from './flickr.service';
5 +
6 @NgModule({
7   imports: [
8     CommonModule
9   ]
10 })
11 export class ServicesModule {
12   static forRoot(): ModuleWithProviders {
13     return {
14       ngModule: ServicesModule,
15       providers: [
16 +       FlickrService
17     ]
18   };
19 }
20 }
21
```

If you run the test script now, you will receive the following error:



Edit the app.module.ts file.

Add imports for the HttpModule and JsonpModule, as show below:

```
1 import { BrowserModule, Title } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 +import { HttpClientModule, JsonpModule } from '@angular/http';
4
5 import './rxjs-extensions';
6
7 import { ServicesModule } from './services/services.module';
8 import { SharedModule } from './shared/shared.module';
9
10 import { AppRoutingModule } from './app-routing.module';
11 import { AppComponent } from './app.component';
12 import { HomeComponent } from './home/home.component';
13 import { ErrorComponent } from './error/error.component';
14 import { NotFoundComponent } from './not-found/not-found.component';
15
16 @NgModule({
17   declarations: [
18     AppComponent,
19     HomeComponent,
20     ErrorComponent,
21     NotFoundComponent
22   ],
23   imports: [
24     BrowserModule,
25 +   HttpClientModule,
26 +   JsonpModule,
27     ServicesModule.forRoot(),
28     SharedModule.forRoot(),
29     SharedModule,
30     AppRoutingModule
31   ],
32   providers: [
33     Title
34   ],
35   bootstrap: [AppComponent]
36 })
37 export class AppModule { }
38 |
```

Add Modularized Camp Site Features

Sunday, August 6, 2017 8:07 AM

You can organize your camp site into self-contained modules that's can be redistributed to your fellow campers for use in their sites.

You could do something like the following:

- Add a camp photos search feature using the Flickr API services.
- Add a site destination lookup feature using the Google Maps services.
- Add a movie trivia feature for game nights using the SWAPI (Star Wars API) services.

Walkthrough adding a flickr module

Edit the **app.module.ts** file.

Add the **ServiceModule** and **SharedModule** to the imports section and be sure to include the **.forRoot()** syntax as show below:

```
1 import { BrowserModule, Title } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import './rxjs-extensions';
5 +import { ServicesModule } from './services/services.module';
6 +import { SharedModule } from './shared/shared.module';
7
8 import { AppRoutingModule } from './app-routing.module';
9 import { AppComponent } from './app.component';
10 import { HomeComponent } from './home/home.component';
11 import { ErrorComponent } from './error/error.component';
12 import { NotFoundComponent } from './not-found/not-found.component';
13
14 @NgModule({
15   declarations: [
16     AppComponent,
17     HomeComponent,
18     ErrorComponent,
19     NotFoundComponent
20   ],
21   imports: [
22     BrowserModule,
23 +   ServicesModule.forRoot(),
24 +   SharedModule.forRoot(),
25     AppRoutingModule
26   ],
27   providers: [
28     Title
29   ],
30   bootstrap: [AppComponent]
31 })
32 export class AppModule { }
```

Generate a new flicker service under the services folder and add it to the services module.

```
ng g s services\flickr
```

```
C:\srcCC\camp-comfy>ng g s services\flickr
installing service
  create src\app\services\flickr.service.spec.ts
  create src\app\services\flickr.service.ts
WARNING Service is generated but not provided, it must be provided to be used
```

Edit the new **flickr.service.ts** file, and replace its contents with the following:

```
import { Injectable } from '@angular/core';
import { Http, Response, RequestOptions, Headers, URLSearchParams } from '@angular/http';

import { Observable } from 'rxjs/Observable';

@Injectable()
export class FlickrService {

  constructor(
    private http: Http
  ) { }

  baseUrl: string = 'https://api.flickr.com/services/rest';

  search(searchText: string): Observable<any> {
    console.log(searchText);

    const params = new URLSearchParams();
    params.set('method', 'flickr.photos.search');
    params.set('format', 'json');
    // Replace the following with your flicker API key
    params.set('api_key', '53286431b01976732160ae74b1c81a8b');
    params.set('action', 'opensearch');
    params.set('text', searchText);
    params.set('per_page', '25');
    params.set('media', 'photos');
    params.set('content_type', '1');
    params.set('format', 'json');
    params.set('nojsoncallback', '1');

    const requestOptions = new RequestOptions();
    requestOptions.search = params;

    return this.http.get(this.baseUrl, requestOptions)
      .map(this.extractData)
      .catch(this.handleError);
  }

  private extractData(res: Response): any[] {

```

```

const photos: any[] = [];

const respPhotos = res.json().photos.photo;

for (let i = 0; i < respPhotos.length; i++) {

  const farm = respPhotos[i].farm;
  const server = respPhotos[i].server;
  const id = respPhotos[i].id;
  const secret = respPhotos[i].secret;
  const title = respPhotos[i].title;

  const photoUrl = `https://farm${farm}.staticflickr.com/${server}/${id}_${secret}.jpg`;

  let titleText = title;

  const pos = title.indexOf('#');

  if (pos > 0) {
    titleText = title.substring(0, pos - 1);
  }

  photos.push({ url: photoUrl, title: titleText });
}

return photos;
}

private handleError(error: Response | any) {

  // In a real world app, you might use a remote logging infrastructure
  let errMsg: string;

  if (error instanceof Response) {
    const body = error.json() || "";
    const err = body.error || JSON.stringify(body);
    errMsg = `${error.status} - ${error.statusText || ""} ${err}`;
  }
  else {
    errMsg = error.message ? error.message : error.toString();
  }

  console.error(errMsg);

  return Observable.throw(errMsg);
}
}

```

Edit the **services.module.ts** file.

Add the **FlickrService** to the providers and import the service as show below:

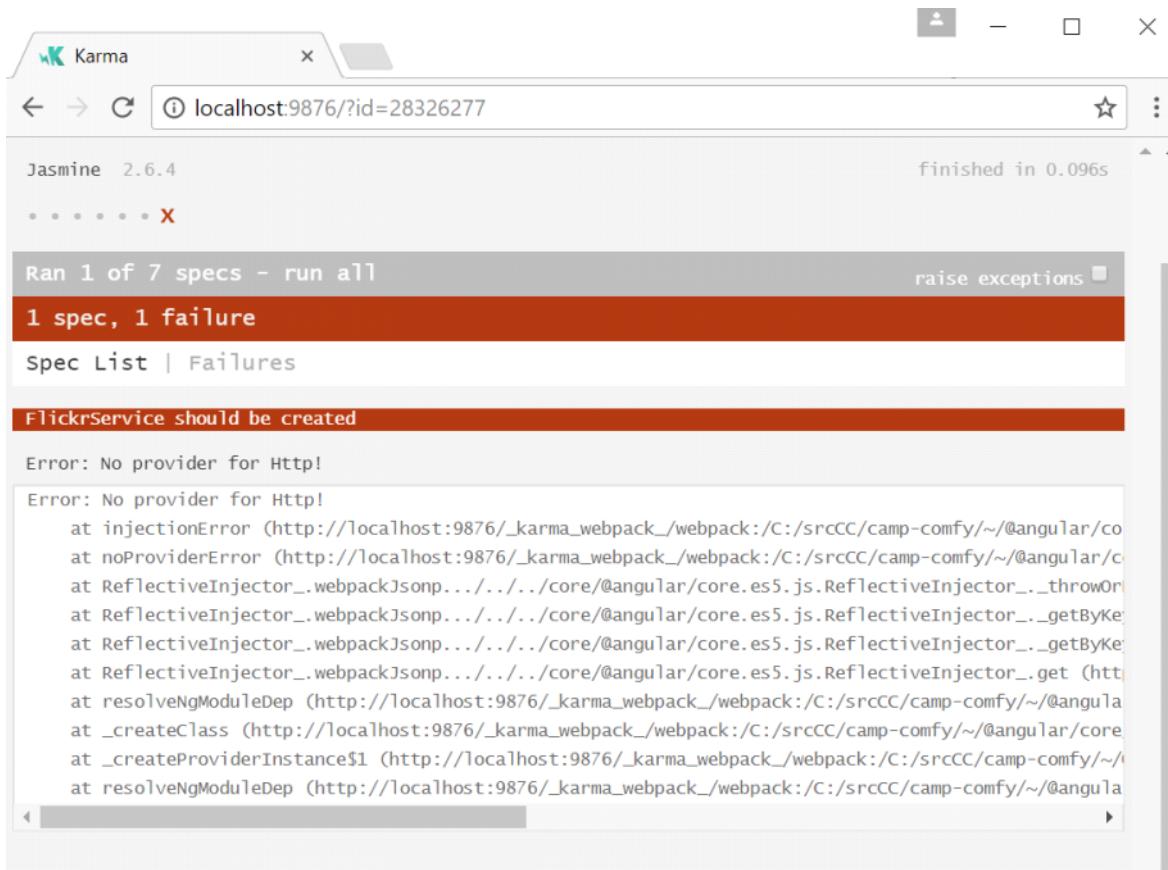
```
1 import { NgModule, ModuleWithProviders } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 +import { FlickrService } from './flickr.service';
5 +
6 @NgModule({
7   imports: [
8     CommonModule
9   ]
10 })
11 export class ServicesModule {
12   static forRoot(): ModuleWithProviders {
13     return {
14       ngModule: ServicesModule,
15       providers: [
16 +       FlickrService
17       ]
18     };
19   }
20 }
21
```

Edit the **services.module.ts** file.

Add a provider entry for the **FlickrService** and modify the code as shown below:

```
1 import { NgModule, ModuleWithProviders } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 +import { FlickrService } from './flicker.service';
5 +
6 @NgModule({
7   imports: [
8     CommonModule
9   ]
10 })
11 export class ServicesModule {
12   static forRoot(): ModuleWithProviders {
13     return {
14       ngModule: ServicesModule,
15       providers: [
16 +       FlickrService
17       ]
18     };
19   }
20 }
21
```

If you run the test script now, you will receive the following error:



Edit the **app.module.ts** file.

Add imports for the **HttpModule** and **JsonpModule**, as show below:

```
1 import { BrowserModule, Title } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 +import { HttpClientModule, JsonpModule } from '@angular/http';
4
5 import './rxjs-extensions';
6
7 import { ServicesModule } from './services/services.module';
8 import { SharedModule } from './shared/shared.module';
9
10 import { AppRoutingModule } from './app-routing.module';
11 import { AppComponent } from './app.component';
12 import { HomeComponent } from './home/home.component';
13 import { ErrorComponent } from './error/error.component';
14 import { NotFoundComponent } from './not-found/not-found.component';
15
16 @NgModule({
17   declarations: [
18     AppComponent,
19     HomeComponent,
20     ErrorComponent,
21     NotFoundComponent
22   ],
23   imports: [
24     BrowserModule,
25 +   HttpClientModule,
26 +   JsonpModule,
27     ServicesModule.forRoot(),
28     SharedModule.forRoot(),
29     SharedModule,
30     AppRoutingModule
31   ],
32   providers: [
33     Title
34   ],
35   bootstrap: [AppComponent]
36 })
37 export class AppModule { }
38 |
```

Edit the **flickr.service.spec.ts** file.

Add imports for the **HttpClientModule** and **JsonpModule** modules as shown below:

```
1 import { TestBed, inject } from '@angular/core/testing';
2 +import { HttpClientModule, JsonpModule } from '@angular/http';
3
4 import { FlickrService } from './flickr.service';
5
6 describe('FlickrService', () => {
7   beforeEach(() => {
8     TestBed.configureTestingModule({
9 +     imports: [
10 +       HttpClientModule,
11 +       JsonpModule
12 +     ],
13 +     providers: [FlickrService]
14   });
15 });
16
17 it('should be created', inject([FlickrService], (service: FlickrService) => {
18   expect(service).toBeTruthy();
19 }));
20 });
21
```

Add Commonly Desired Camp Site Views

Sunday, July 23, 2017 12:49 PM

Most camp sites have a commonly required (desired) views.

These view should usually be loaded when the application starts.

Open a command line window.

Add a home view.

`ng g c home`

```
C:\srcCC\camp-comfy>ng g c home
installing component
  create src\app\home\home.component.scss
  create src\app\home\home.component.html
  create src\app\home\home.component.spec.ts
  create src\app\home\home.component.ts
  update src\app\app.module.ts
```

Replace the content in the html file with something like the following:

```
1 <h1>Welcome to {{title}}!</h1>
2
3 <p>
4   | We have a lot of fun and educational activities planned for you.
5 </p>
6
7 <p>
8   | We hope you enjoy your time here at Camp Comfy.
9 </p>
10
11 <p>
12   | <button type="button" class="btn btn-primary" [routerLink]="['/error']">Error</button>
13 </p>
14
15 <p>
16   | <button type="button" class="btn btn-primary" [routerLink]="['/not-found']">Not Found</button>
17 </p>
18 |
```

```
<h1>Welcome to {{title}}!</h1>
```

```
<p>
  We have a lot of fun and educational activities planned for you.
</p>
```

```
<p>
  We hope you enjoy your time here at Camp Comfy.
</p>
```

```

<p>
  <button type="button" class="btn btn-primary" [routerLink]=["'/error']>Error</button>
</p>

<p>
  <button type="button" class="btn btn-primary" [routerLink]=["'/not-found']>Not Found</button>
</p>

```

Add an error view.

`ng g c error`

```
C:\srcCC\camp-comfy>ng g c error
installing component
  create src\app\error\error.component.scss
  create src\app\error\error.component.html
  create src\app\error\error.component.spec.ts
  create src\app\error\error.component.ts
  update src\app\app.module.ts
```

Replace the content in the html file with something like the following:

```

1  <h1>Error</h1>
2
3  <p>
4    We're sorry, but something went wrong with our camp site equipment.
5  </p>
6
7  <p>
8    Please try your last camp activity again.
9  </p>
10
11 <p>
12   If the problem persists, please contact your camp counselor for assistance.
13 </p>
14
15 <p>
16   <button type="button" class="btn btn-primary" [routerLink]=["'/home']>Home</button>
17 </p>
```

```

<h1>Error</h1>

<p>
We're sorry, but something went wrong with our camp site equipment.
</p>

<p>
Please try your last camp activity again.
</p>

<p>
If the problem persists, please contact your camp counselor for assistance.
</p>
```

```
<p>
  <button type="button" class="btn btn-primary" [routerLink]=["'/home']>Home</button>
</p>
```

Add a not-found view.

```
ng g c not-found
```

```
C:\srcCC\camp-comfy>ng g c not-found
installing component
  create src\app\not-found\not-found.component.scss
  create src\app\not-found\not-found.component.html
  create src\app\not-found\not-found.component.spec.ts
  create src\app\not-found\not-found.component.ts
  update src\app\app.module.ts
```

Replace the content in the html file with something like the following:

```
1  <h1>Not Found</h1>
2
3  <p>
4    We're sorry, but we are unable to provide the camp site location you requested.
5  </p>
6
7  <p>
8    Please submit a request for a different location.
9  </p>
10
11 <p>
12   <button type="button" class="btn btn-primary" [routerLink]=["'/home']>Home</button>
13 </p>
14 |
```

```
<h1>Not Found</h1>
```

```
<p>
  We're sorry, but we are unable to provide the camp site location you requested.
</p>
```

```
<p>
  Please submit a request for a different location.
</p>
```

```
<p>
  <button type="button" class="btn btn-primary" [routerLink]=["'/home']>Home</button>
</p>
```

Edit the app-routing.module.ts file.

Replace the code with the following:

```

1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3
4 import { HomeComponent } from './home/home.component';
5 import { ErrorComponent } from './error/error.component';
6 import { NotFoundComponent } from './not-found/not-found.component';
7
8 const routes: Routes = [
9 {
10     path: 'home',
11     component: HomeComponent,
12     data: {
13         title: 'Welcome!'
14     }
15 },
16 {
17     path: 'error',
18     component: ErrorComponent,
19     data: {
20         title: 'Error'
21     }
22 },
23 {
24     path: '',
25     redirectTo: 'home',
26     pathMatch: 'full'
27 },
28 {
29     path: '**',
30     component: NotFoundComponent,
31     data: {
32         title: 'Not Found'
33     }
34 },
35 ];
36
37 @NgModule({
38     imports: [RouterModule.forRoot(routes)],
39     exports: [RouterModule]
40 })
41 export class AppRoutingModule { }
42

```

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { HomeComponent } from './home/home.component';
import { ErrorComponent } from './error/error.component';
import { NotFoundComponent } from './not-found/not-found.component';

const routes: Routes = [
{

```

```

path: 'home',
component: HomeComponent,
data: {
  title: 'Welcome!'
}
},
{
  path: 'error',
  component: ErrorComponent,
  data: {
    title: 'Error'
  }
},
{
  path: '',
  redirectTo: 'home',
  pathMatch: 'full'
},
{
  path: '**',
  component: NotFoundComponent,
  data: {
    title: 'Not Found'
  }
},
];

```

```

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }

```

Edit the app.component.html file.

Replace the content with the following:

```

1 | <div class="app-container">
2 |   <div class="spacer-15"></div>
3 |   <div class="page-container">
4 |     <router-outlet></router-outlet>
5 |   </div>
6 | </div>
7 | <div class="spacer-15"></div>
8

```

```

<div class="app-container">
<div class="spacer-15"></div>
<div class="page-container">
  <router-outlet></router-outlet>

```

```
</div>
</div>
<div class="spacer-15"></div>
```

Edit the app.component.scss file.

Add the following styles:

```
1  .app-container {
2    max-width: 1024px;
3    margin: 0px auto;
4  }
5
6  .page-container {
7    margin: 0px 10px;
8  }
9
```

```
.app-container {
  max-width: 1024px;
  margin: 0px auto;
}

.page-container {
  margin: 0px 10px;
}
```

Edit the **styles.scss** file.

Add the following styles:

```
1  /* You can add global styles to this file, and also import other style files */
2  @import 'variables';
3  @import 'bootstrap';
4
5  .spacer-5 {
6      height: 5px;
7  }
8
9  .spacer-10 {
10     height: 10px;
11 }
12
13 .spacer-15 {
14     height: 15px;
15 }
16
17 .spacer-20 {
18     height: 20px;
19 }
20
21 .spacer-25 {
22     height: 25px;
23 }
24
25 .spacer-30 {
26     height: 30px;
27 }
```

```
.spacer-5 {
    height: 5px;
}
```

```
.spacer-10 {
    height: 10px;
}
```

```
.spacer-15 {
    height: 15px;
}
```

```
.spacer-20 {
    height: 20px;
}
```

```
.spacer-25 {
    height: 25px;
}
```

```
.spacer-30 {
    height: 30px;
}
```

Fix the broken unit tests.

Edit the home.component.spec.ts file.

Add the new code as show below:

```
1 import { async, ComponentFixture, TestBed } from '@angular/core/testing';
2 +import { RouterTestingModule } from '@angular/router/testing';
3 +import { DebugElement } from '@angular/core';
4 +import { By } from '@angular/platform-browser';
5
6 import { HomeComponent } from './home.component';
7
8 describe('HomeComponent', () => {
9   let component: HomeComponent;
10  let fixture: ComponentFixture<HomeComponent>;
11
12  beforeEach(async(() => {
13    TestBed.configureTestingModule({
14 +    declarations: [ HomeComponent ],
15 +    imports: [
16 +      RouterTestingModule.withRoutes([
17 +        {
18 +          path: 'error',
19 +          component: HomeComponent
20 +        }
21 +      ])
22 +    ],
23    })
24    .compileComponents();
25  }));

```

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { DebugElement } from '@angular/core';
import { By } from '@angular/platform-browser';

import { HomeComponent } from './home.component';

describe('HomeComponent', () => {
  let component: HomeComponent;
  let fixture: ComponentFixture<HomeComponent>

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ HomeComponent ],
      imports: [
        RouterTestingModule.withRoutes([
          {
            path: 'error',

```

```

        component: HomeComponent
    }
])
],
})
.compileComponents();
})));

```

Edit the error.component.spec.ts file.

Add the new code as show below:

```

1 import { async, ComponentFixture, TestBed } from '@angular/core/testing';
2 +import { RouterTestingModule } from '@angular/router/testing';
3 +import { DebugElement } from '@angular/core';
4 +import { By } from '@angular/platform-browser';
5
6 import { ErrorComponent } from './error.component';
7
8 describe('ErrorComponent', () => {
9     let component: ErrorComponent;
10    let fixture: ComponentFixture<ErrorComponent>;
11
12    beforeEach(async(() => {
13        TestBed.configureTestingModule({
14 +            declarations: [ErrorComponent],
15 +            imports: [
16 +                RouterTestingModule.withRoutes([
17 +                    {
18 +                        path: 'error',
19 +                        component: ErrorComponent
20 +                    }
21 +                ])
22 +            ],
23    })
24        .compileComponents();
25    }));

```

```

import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { DebugElement } from '@angular/core';
import { By } from '@angular/platform-browser';

import { ErrorComponent } from './error.component';

describe('ErrorComponent', () => {
    let component: ErrorComponent;
    let fixture: ComponentFixture<ErrorComponent>

    beforeEach(async(() => {
        TestBed.configureTestingModule({

```

```

declarations: [ErrorComponent],
imports: [
  RouterTestingModule.withRoutes([
    {
      path: 'error',
      component: ErrorComponent
    }
  ])
],
})
.compileComponents();
})));

```

Edit the not-found.component.spec.ts file.

Add the new code as show below:

```

1 import { async, ComponentFixture, TestBed } from '@angular/core/testing';
2 +import { RouterTestingModule } from '@angular/router/testing';
3 +import { DebugElement } from '@angular/core';
4 +import { By } from '@angular/platform-browser';
5
6 import { NotFoundComponent } from './not-found.component';
7
8 describe('NotFoundComponent', () => {
9   let component: NotFoundComponent;
10  let fixture: ComponentFixture<NotFoundComponent>;
11
12  beforeEach(async(() => {
13    TestBed.configureTestingModule({
14 +      declarations: [ NotFoundComponent ],
15 +      imports: [
16 +        RouterTestingModule.withRoutes([
17 +          {
18 +            path: 'not-found',
19 +            component: NotFoundComponent
20 +          }
21 +        ])
22 +      ],
23    })
24    .compileComponents();
25  }));

```

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { RouterTestingModule } from '@angular/router/testing';
```

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

```
import { By } from '@angular/platform-browser';
```

```
import { NotFoundComponent } from './not-found.component';
```

```
describe('NotFoundComponent', () => {
  let component: NotFoundComponent;
  let fixture: ComponentFixture<NotFoundComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ NotFoundComponent ],
      imports: [
        RouterTestingModule.withRoutes([
          {
            path: 'not-found',
            component: NotFoundComponent
          }
        ])
      ],
    })
    .compileComponents();
  }));
});
```

Design The Layout Of Your Camp Site

Sunday, August 6, 2017 8:08 AM

Be responsive to the diversity of your camper's equipment.

Design the camp site container specifications that set the boundaries of your camp site.

Design the camp site header (the first thing campers see).

Design the camp site footer (the last thing campers see)

Design various camp site content styles to suit your campers personal sense of esthetics.

Walkthrough Site Layout

Add Page Header Component

ng g c shared\page-header

```
C:\srcCC\camp-comfy>ng g c shared\page-header
installing component
  create src\app\shared\page-header\page-header.component.scss
  create src\app\shared\page-header\page-header.component.html
  create src\app\shared\page-header\page-header.component.spec.ts
  create src\app\shared\page-header\page-header.component.ts
  update src\app\shared\shared.module.ts
```

Edit the **page-header.component.html** file.

Replace the content with the following:

```
1  <div class="page-header">
2    |  <h1 class="app-name">Camp Comfy</h1>
3  </div>
4
```

```
<div class="page-header">
  <h1 class="app-name">Camp Comfy</h1>
</div>
```

Edit the **page-header.component.scss** file.

Replace the content with the following:

```
1 .page-header {  
2     padding: 15px;  
3     color: #F2E6DB;  
4     background-color: #99593D;  
5     border-top-left-radius: 5px;  
6     border-top-right-radius: 5px;  
7 }  
8  
9 h1.app-name {  
10    margin: 0;  
11    padding: 0;  
12 }  
13 |
```

```
.page-header {  
  padding: 15px;  
  color: #F2E6DB;  
  background-color: #99593D;  
  border-top-left-radius: 5px;  
  border-top-right-radius: 5px;  
}  
  
h1.app-name {  
  margin: 0;  
  padding: 0;  
}
```

Add Page Footer Component

```
ng g c shared\page-footer
```

```
C:\srcCC\camp-comfy>ng g c shared\page-footer  
installing component  
  create src\app\shared\page-footer\page-footer.component.scss  
  create src\app\shared\page-footer\page-footer.component.html  
  create src\app\shared\page-footer\page-footer.component.spec.ts  
  create src\app\shared\page-footer\page-footer.component.ts  
  update src\app\shared\shared.module.ts
```

Edit the **page-footer.component.html** file.

Replace the content with the following:

```
1 <footer>  
2   Comfy Coder &copy; 2017  
3 </footer>  
4 |
```

```
<footer>  
  Comfy Coder &copy; 2017  
</footer>
```

Edit the **page-footer.component.scss** file.

Replace the content with the following:

```
1  footer {
2      padding: 15px;
3      margin: 0;
4      color: #F2E6DB;
5      background-color: #5E6251;
6      border-bottom-left-radius: 5px;
7      border-bottom-right-radius: 5px;
8  }
9
```

```
footer {
  padding: 15px;
  margin: 0;
  color: #F2E6DB;
  background-color: #5E6251;
  border-bottom-left-radius: 5px;
  border-bottom-right-radius: 5px;
}
```

Edit the **shared.module.ts** file.

Modify the code as show in the following:

```

1 import { NgModule, ModuleWithProviders } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 +import { PageHeaderComponent } from './page-header/page-header.component';
5 +import { PageFooterComponent } from './page-footer/page-footer.component';
6 +
7 @NgModule({
8   imports: [
9 +   CommonModule,
10 ],
11 declarations: [
12 +   PageHeaderComponent,
13 +   PageFooterComponent
14 ],
15 exports: [
16 +   PageHeaderComponent,
17 +   PageFooterComponent
18 ]
19 })
20 export class SharedModule {
21   static forRoot(): ModuleWithProviders {
22     return {
23       ngModule: SharedModule,
24       providers: [
25
26     ]
27   };
28 }
29 }
30

```

Edit the **app.component.html** file.

Replace its contents with the following:

```

1 +<div class="spacer-15"></div>
2 <div class="app-container">
3 +  <div class="app-inner-container">
4 +    <cc-page-header></cc-page-header>
5      <div class="spacer-15"></div>
6      <div class="page-container">
7        <router-outlet></router-outlet>
8 +
9 +    </div>
10 +   <div class="spacer-15"></div>
11 +   <cc-page-footer></cc-page-footer>
12   </div>
13 <div class="spacer-15"></div>
14

```

```

<div class="spacer-15"></div>
<div class="app-container">
  <div class="app-inner-container">
    <cc-page-header></cc-page-header>
    <div class="spacer-15"></div>
    <div class="page-container">

```

```
<router-outlet></router-outlet>
</div>
<div class="spacer-15"></div>
<cc-page-footer></cc-page-footer>
</div>
</div>
<div class="spacer-15"></div>
```

Edit the **app.component.scss** file.

Replace its contents with the following:

```
1  .app-container {
2 +   max-width: 768px;
3 +   margin: 0px auto;
4 +   padding: 0 15px;
5  }
6
7 + .app-inner-container {
8 +   border: 1px solid #9D827C;
9 +   -moz-border-radius: 6px;
10 +  border-radius: 6px;
11 +  box-shadow: 0px 2px 35px 2px rgba(0, 0, 0, 0.25);
12 +  background-color: #F2E6DB;
13 +}
14 +
15 .page-container {
16 +  margin: 0px 15px;
17 }
18
```

```
.app-container {
  max-width: 768px;
  margin: 0px auto;
  padding: 0 15px;
}

.app-inner-container {
  border: 1px solid #9D827C;
  -moz-border-radius: 6px;
  border-radius: 6px;
  box-shadow: 0px 2px 35px 2px rgba(0, 0, 0, 0.25);
  background-color: #F2E6DB;
}

.page-container {
  margin: 0px 15px;
}
```

Edit the **styles.scss** file.

Add the following new style:

```
29 + body {  
30 +   color: #534134;  
31 +   background-color: #CCBAAA;  
32 + } [  
33 + ]
```

```
body {  
  color:#534134;  
  background-color: #CCBAAA;  
}
```

Edit the **_variables.scss** file.

Change the following style:

```
1 + $brand-primary: #6E0D11;  
2
```

```
$brand-primary: #6E0D11;
```

Edit the **app.component.spec.ts** file.

Modify the code as shown in the following:

```

1 import { TestBed, async } from '@angular/core/testing';
2 import { RouterTestingModule } from '@angular/router/testing';
3
4 import './rxjs-extensions';
5
6 import { AppComponent } from './app.component';
7 +import { SharedModule } from './shared/shared.module';
8
9 describe('AppComponent', () => {
10   beforeEach(async(() => {
11     TestBed.configureTestingModule({
12       imports: [
13 +         RouterTestingModule,
14 +         SharedModule
15       ],
16       declarations: [
17         AppComponent
18       ],
19     }).compileComponents();
20   }));
21
22   it('should create the app', async(() => {
23     const fixture = TestBed.createComponent(AppComponent);
24     const app = fixture.debugElement.componentInstance;
25     expect(app).toBeTruthy();
26   }));
27 });
28

```

Note that you must import the **SharedModule** so that the app component test can find the new page header and footer components.

Edit the **app.po.ts** file.

Modify the code as show by the following:

```

1 import { browser, by, element } from 'protractor';
2
3 export class CampComfyPage {
4   navigateTo() {
5     return browser.get('/');
6   }
7
8   getParagraphText() {
9 +   return element(by.css('cc-root h2')).getText();
10 }
11 }
12

```

Feeling Lazy? Load Those Features On Demand

Saturday, July 22, 2017 9:01 AM

Feature modules allow you to group and load related application features on demand, which helps decrease the application startup time.

No need to load everything at once, you should only load what the user expects to see when the app first appears.

Lazy loaded feature modules only get loaded when a user navigates to the feature's route.

Implement Feature Modules

<https://angular.io/guide/ngmodule#feature-modules>

Implement Lazy Loading

<https://angular.io/guide/ngmodule#lazy-loading-modules-with-the-router>

Walkthrough implementing flickr search feature

Add Flickr Search Feature Module with Routing

ng g m search-flickr -routing

```
C:\srcCC\camp-comfy>ng g m search-flickr -routing
installing module
  create src\app\search-flickr\search-flickr-routing.module.ts
  create src\app\search-flickr\search-flickr.module.ts
WARNING Module is generated but not provided, it must be provided to be used
```

Edit the **app.module.ts** file.

Modify the code to import **FormsModule**, as show in the following:

```
1 import { BrowserModule, Title } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 +import { FormsModule } from '@angular/forms';
4 import { HttpClientModule, JsonpModule } from '@angular/http';
5
6 import './rxjs-extensions';
7
8 import { ServicesModule } from './services/services.module';
9 import { SharedModule } from './shared/shared.module';
10
11 import { AppRoutingModule } from './app-routing.module';
12 import { AppComponent } from './app.component';
13 import { HomeComponent } from './home/home.component';
14 import { ErrorComponent } from './error/error.component';
15 import { NotFoundComponent } from './not-found/not-found.component';
16
17 @NgModule({
18   declarations: [
```

```

16
17  @NgModule({
18    declarations: [
19      AppComponent,
20      HomeComponent,
21      ErrorComponent,
22      NotFoundComponent
23    ],
24    imports: [
25      BrowserModule,
26 +   FormsModule,
27      HttpClientModule,
28      JsonpModule,
29      ServicesModule.forRoot(),
30      SharedModule.forRoot(),
31      SharedModule,
32      AppRoutingModule
33    ],
34    providers: [
35      Title
36    ],
37    bootstrap: [AppComponent]
38  })
39  export class AppModule { }
40

```

Edit the **search-flickr.module.ts** file.

Modify the code to import **FormsModule**, as show in the following:

```

1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3 +import { FormsModule } from '@angular/forms';
4
5  import { SearchFlickrRoutingModule } from './search-flickr-routing.module';
6  import { SearchFlickrComponent } from './search-flickr.component';
7
8  @NgModule({
9    imports: [
10      CommonModule,
11 +    FormsModule,
12      SearchFlickrRoutingModule
13    ],
14 +  declarations: [
15 +    SearchFlickrComponent
16 +  ]
17  })
18  export class SearchFlickrModule { }
19

```

Edit the **app-routing.module.ts** file.

Add the following to the **Routes** array:

```
16  {
17 +   path: 'search',
18 +   loadChildren: './search-flickr/search-flickr.module#SearchFlickrModule'
19 + },
```

This syntax tells the router to lazy load the search flickr view when the user navigates to it.

Edit the **search-flickr-routing.module.ts** file.

Add the **SearchFlickrComponent** as the default entry in the Routes array, as show below.

```
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3
4 +import { SearchFlickrComponent } from './search-flickr.component';
5 +
6 +const routes: Routes = [
7 +  {
8 +    path: '',
9 +    component: SearchFlickrComponent
10+  }
11+];
12
13 @NgModule({
14   imports: [RouterModule.forChild(routes)],
15   exports: [RouterModule]
16 })
17 export class SearchFlickrRoutingModule { }
```

Add Flickr Search Feature Component View

ng g c search-flickr

```
C:\srcCC\camp-comfy>ng g c search-flickr
installing component
  create src\app\search-flickr\search-flickr.component.scss
  create src\app\search-flickr\search-flickr.component.html
  create src\app\search-flickr\search-flickr.component.spec.ts
  create src\app\search-flickr\search-flickr.component.ts
  update src\app\search-flickr\search-flickr.module.ts
```

Edit the **search-flickr.component.html** file.

Replace the contents with the following:

```
<h1>Search Flickr</h1>
<p>
  <button type="button" class="btn btn-primary" [routerLink]="/home">
    <i class="fa fa-chevron-left" aria-hidden="true"></i>
    Back
  </button>
</p>
```

```

<form class="search-form">
  <label class="search-label">Search for Flickr Images:</label>
  <div class="input-group">
    <input [(ngModel)]="searchString" name="search-input" type="text" class="form-control"
      autocomplete="off" (keydown.enter)="search()" />
    <span (click)="search()" class="input-group-addon" id="btnGroupAddon2">
      <i class="fa fa-search" aria-hidden="true"></i>
    </span>
  </div>
</form>
<div class="spacer-15"></div>
<div class="row">
  <div class="col-xl-12" *ngFor="let photo of photos">
    <div class="card">
      <img [src]="photo.url" alt="Avatar" style="width:100%;height:auto;">
      <div class="container">
        <p class="card-text">{{photo.title}}</p>
      </div>
    </div>
  </div>
</div>

```

Edit the **search-flickr.component.scss** file.

Add the following styles:

```

.card {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  margin: 10px auto;
  text-align: center;
}

.container {
  padding: 10px;
}

```

Edit the **search-flickr.component.ts** file.

Replace all of the code with the following:

```

import { Component, OnInit, OnDestroy } from '@angular/core';
import { Router } from '@angular/router';
import { Observable } from 'rxjs/Observable';

import { FlickrService } from './services/flickr.service';

@Component({
  selector: 'cc-search-flickr',
  templateUrl: './search-flickr.component.html',
  styleUrls: ['./search-flickr.component.scss']
})

```

```

export class SearchFlickrComponent implements OnInit, OnDestroy {

  searchString: string = '';
  photos: any[] = [];

  constructor(
    private flickrService: FlickrService,
    private router: Router
  ) {}

  ngOnInit() {
  }

  ngOnDestroy() {
  }

  search() {
    this.photos = [];

    if (this.searchString.length === 0) {
      alert('Please first enter a search query.');
      return;
    }

    this.flickrService.search(this.searchString).subscribe(
      data => {
        if (data.length === 0) {
          console.log('no photos matched your search query.');
        } else {
          console.log(data);
          this.photos = data;
          console.log(this.photos);
        }
      },
      err => {
        console.log(err);
        this.router.navigate(['/error']);
      },
      () => {
        console.log('Flickr search complete');
      }
    );
  }
}

```

Edit the **search-flickr.component.spec.ts** file.

Replace all of the code with the following:

```

import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/http';
import { RouterTestingModule } from '@angular/router/testing';

import { SearchFlickrComponent } from './search-flickr.component';

import { FlickrService } from '../services/flickr.service';

describe('SearchFlickrComponent', () => {
  let component: SearchFlickrComponent;
  let fixture: ComponentFixture<SearchFlickrComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      imports: [
        FormsModule,
        HttpClientModule,
        RouterTestingModule.withRoutes([
          {
            path: 'search',
            component: SearchFlickrComponent
          }
        ])
      ],
      declarations: [
        SearchFlickrComponent
      ],
      providers: [
        FlickrService
      ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(SearchFlickrComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should be created', () => {
    expect(component).toBeTruthy();
  });
});

```

Name Your Camp Site Locations

Saturday, July 29, 2017 2:33 PM

You can mark the carious view locations while browsing Camp Comfy.

Add a data section to each of you cam site routes with a title value.

Do something like the following:

```
8  const routes: Routes = [
9    {
10      path: 'home',
11      component: HomeComponent,
12      data: {
13        title: 'Welcome!'
14      }
15    },
16  ]
```

Update Your Current Camp Site Location

Sunday, August 6, 2017 8:15 AM

Angular provides a Title service as part of the Browser platform. The Title service provides an API for getting and setting the current HTML document title.

<https://angular.io/guide/set-document-title>

Add rxjs extensions.

Create a new file in the app folder called:

rxjs-extensions.ts

The purpose of this file is to facilitate the importing of only specific exports from the RXJS library and make them globally available to all other modules.

Add the following commonly needed RXJS module imports to this file:

```
1 import 'rxjs/add/operator/catch';
2 import 'rxjs/add/operator/do';
3 import 'rxjs/add/operator/map';
4 import 'rxjs/add/operator/toPromise';
5 import 'rxjs/add/observable/of';
6 import 'rxjs/add/observable/throw';
7 import 'rxjs/add/operator/filter';
8
```

```
import 'rxjs/add/operator/catch';
import 'rxjs/add/operator/do';
import 'rxjs/add/operator/map';
import 'rxjs/add/operator/toPromise';
import 'rxjs/add/observable/of';
import 'rxjs/add/observable/throw';
import 'rxjs/add/operator/filter';
```

Edit the app.module.ts file.

Make the following changes:

```

1 +import { BrowserModule, Title } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 +
4 +import './rxjs-extensions';
5
6 import { AppRoutingModule } from './app-routing.module';
7 import { AppComponent } from './app.component';
8 import { HomeComponent } from './home/home.component';
9 import { ErrorComponent } from './error/error.component';
10 import { NotFoundComponent } from './not-found/not-found.component';
11
12 @NgModule({
13   declarations: [
14     AppComponent,
15     HomeComponent,
16     ErrorComponent,
17     NotFoundComponent
18   ],
19   imports: [
20     BrowserModule,
21     AppRoutingModule
22   ],
23 + providers: [
24 +   Title
25 + ],
26   bootstrap: [AppComponent]
27 })
28 export class AppModule { }
29

```

Or simply replace the code in the file with the following:

```

import { BrowserModule, Title } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import './rxjs-extensions';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HomeComponent } from './home/home.component';
import { ErrorComponent } from './error/error.component';
import { NotFoundComponent } from './not-found/not-found.component';

@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    ErrorComponent,
    NotFoundComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [
    Title
  ]
})
export class AppModule { }

```

```

],
bootstrap: [AppComponent]
})
export class AppModule { }

```

Add a means to set the browser title.

Edit the app.component.ts module.

Replace the code with the following:

```

1 import { Component, OnInit, OnDestroy } from '@angular/core';
2 import { Title } from '@angular/platform-browser';
3 import { Router, NavigationStart, NavigationEnd, ActivatedRouteSnapshot } from '@angular/router';
4 import { Observable } from 'rxjs/Observable';
5
6 @Component({
7   selector: 'cc-root',
8   templateUrl: './app.component.html',
9   styleUrls: ['./app.component.scss']
10 })
11 export class AppComponent implements OnInit, OnDestroy {
12
13   title: Observable<string>;
14   titleSubscription: any;
15
16   public constructor(
17     private router: Router,
18     private titleService: Title
19   ) { }
20
21   ngOnInit() {
22
23     this.title = this.router.events
24       .filter((event) => event instanceof NavigationEnd)
25       .map(() => {
26         const routeSnapshot = this.router.routerState.snapshot.root;
27         return this.getDeepestTitle(routeSnapshot);
28       });
29
30     this.titleSubscription = this.title.subscribe((title) => {
31       const titleComplete = title && title.length > 0 ? 'Camp Comfy - ' + title : 'Camp Comfy';
32       this.titleService.setTitle(titleComplete);
33     });
34   }
35
36   ngOnDestroy() {
37     if (this.titleSubscription) {
38       this.titleSubscription.unsubscribe();
39     }
40   }
41
42   private getDeepestTitle(routeSnapshot: ActivatedRouteSnapshot) {
43     const title = routeSnapshot.firstChild && routeSnapshot.firstChild.data &&
44       routeSnapshot.firstChild.data.title ? routeSnapshot.firstChild.data.title : '';
45     return title;
46   }
47 }

```

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

```

import { Title } from '@angular/platform-browser';
import { Router, NavigationStart, NavigationEnd, ActivatedRouteSnapshot } from '@angular/router';
import { Observable } from 'rxjs/Observable';

@Component({
  selector: 'cc-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit, OnDestroy {

  title: Observable<string>;
  titleSubscription: any;

  public constructor(
    private router: Router,
    private titleService: Title
  ) {}

  ngOnInit() {

    this.title = this.router.events
      .filter((event) => event instanceof NavigationEnd)
      .map(() => {
        const routeSnapshot = this.router.routerState.snapshot.root;
        return this.getDeepestTitle(routeSnapshot);
      });

    this.titleSubscription = this.title.subscribe((title) => {
      const titleComplete = title && title.length > 0 ? 'Camp Comfy - ' + title : 'Camp Comfy';
      this.titleService.setTitle(titleComplete);
    });
  }

  ngOnDestroy() {
    if (this.titleSubscription) {
      this.titleSubscription.unsubscribe();
    }
  }

  private getDeepestTitle(routeSnapshot: ActivatedRouteSnapshot) {
    const title = routeSnapshot.firstChild && routeSnapshot.firstChild.firstChild.data &&
      routeSnapshot.firstChild.data.title ? routeSnapshot.firstChild.data.title : '';
    return title;
  }
}

```

Edit the app.component.spec.ts file.

Add a an import statement for the rxjs-extensions.

```
1 import { TestBed, async } from '@angular/core/testing';
2 import { RouterTestingModule } from '@angular/router/testing';
3
4 +import './rxjs-extensions';
5 +
6 import { AppComponent } from './app.component';
7
8 describe('AppComponent', () => {
9   beforeEach(async(() => {
10     TestBed.configureTestingModule({
11       imports: [
12         RouterTestingModule
13       ],
14       declarations: [
15         AppComponent
16       ],
17     }).compileComponents();
18   }));
19
20   it('should create the app', async(() => {
21     const fixture = TestBed.createComponent(AppComponent);
22     const app = fixture.debugElement.componentInstance;
23     expect(app).toBeTruthy();
24   }));
25 });
26
```

```
import './rxjs-extensions';
```

Provide Directives To Guide Campers On The Trail

Saturday, July 22, 2017 9:02 AM

Issue directives to help your campers correctly fill out their data requisition forms.

Examples:

Filter out invalid characters as they are entered into the form fields:

Names, Titles, Numbers, Decimals, Currency, Dates

Fix up and format text input values for display readability

Numbers, Decimals, Currency, Percentages, Dates

Let Everyone Know When You're Busy

Saturday, July 22, 2017 9:04 AM

Your campers are always eager to explore the features of your campsite.

However, in this day and age of immediate gratification, a camper's need for immediate feedback is high. They tend to get a little impatient and want to know what's happening at the camp site.

You should let them know when you're busy and that progress is being made to service their requests.

Load a Spinner npm Package

Install spin.js.

npm install --save spin.js

```
C:\srcCC\camp-comfy>npm install --save spin.js
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ spin.js@2.3.2
added 115 packages in 19.038s
```

Edit the **.angular-cli.json** file.

Add **spin.js** to the scripts section as shown in the following:

```
24      "scripts": [
25        "../node_modules/jquery/dist/jquery.slim.min.js",
26        "../node_modules/tether/dist/js/tether.min.js",
27 +       "../node_modules/bootstrap/dist/js/bootstrap.min.js",
28 +       "../node_modules/spin.js/spin.min.js"
29    ],
```

Edit the **wallaby.js** file.

Add the **spin.js** file to the files section as shown in the following:

```

39  return {
40    files: [
41      { pattern: 'node_modules/jquery/dist/jquery.slim.min.js', load: true },
42      { pattern: 'node_modules/tether/dist/js/tether.min.js', load: true },
43      { pattern: 'node_modules/bootstrap/dist/js/bootstrap.min.js', load: true },
44 +     { pattern: 'node_modules/spin.js/spin.min.js', load: true },
45     { pattern: 'src/**/*.ts', load: false },
46     { pattern: 'src/**/*.d.ts', ignore: true },
47     { pattern: 'src/**/*.css', load: false },
48     { pattern: 'src/**/*.less', load: false },
49     { pattern: 'src/**/*.scss', load: false },
50     { pattern: 'src/**/*.sass', load: false },
51     { pattern: 'src/**/*.styl', load: false },
52     { pattern: 'src/**/*.html', load: false },
53     { pattern: 'src/**/*.json', load: false },
54     { pattern: 'src/**/*spec.ts', ignore: true }
55   ],

```

Add A Spinner Service

Generate a new spinner service.

ng g s services\spinner

Edit the **spinner.service.ts** file.

Replace the entire contents with the following:

```

import { Injectable } from '@angular/core';
import { Subject } from 'rxjs/Subject';
import { Observable } from 'rxjs/Observable';

@Injectable()
export class SpinnerService {

  private spinnerSubject = new Subject<boolean>();
  public spinnerObservable: Observable<boolean>;

  constructor() {
    this.spinnerObservable = this.spinnerSubject.asObservable();
  }

  show() {
    this.spinnerSubject.next(true);
  }

  hide() {
    this.spinnerSubject.next(false);
  }
}

```

Edit the **spinner.service.spec.ts** file.

Replace the entire contents with the following:

```
import { TestBed, inject } from '@angular/core/testing';
import { SpinnerService } from './spinner.service';

describe('SpinnerService', () => {
  beforeEach(() => {
    TestBed.configureTestingModule({
      providers: [SpinnerService]
    });
  });

  it('should create spinner', inject([SpinnerService], (service: SpinnerService) => {
    expect(service).toBeTruthy();
  }));
});
```

Edit the **services.module.ts** file.

Modify the contents indicated by the following:

```
1 import { NgModule, ModuleWithProviders } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 import { FlickrService } from './flickr.service';
5 +import { SpinnerService } from './spinner.service';
6
7 @NgModule({
8   imports: [
9     CommonModule
10   ]
11 })
12 export class ServicesModule {
13   static forRoot(): ModuleWithProviders {
14     return {
15       ngModule: ServicesModule,
16       providers: [
17 +         FlickrService,
18 +         SpinnerService
19       ]
20     };
21   }
22 }
23
```

Add A Busy Indicator

Generate a new spinner shared component.

```
ng g c shared\spinner
```

Edit the **spinner.component.html** file.

Replace the entire contents with the following:

```
<div class="ng2-spinner" [style.display]="show == true ? 'inherit' : 'none'"></div>
```

Edit the **spinner.component.scss** file.

Add the following style:

```
.ng2-spinner {  
  position: fixed;  
  top: 0;  
  left: 0;  
  right: 0;  
  bottom: 0;  
  width: 100%;  
  height: 100vh;  
  background-color: rgba(0,0,0,0.0);  
  z-index: 100000;  
}
```

Edit the **spinner.component.ts** file.

Replace the entire contents with the following:

```
import { Component, ElementRef, OnInit, OnDestroy, Input } from '@angular/core';  
import { SpinnerService } from '../services/spinner.service';  
import { Subscription } from 'rxjs/Subscription';  
  
declare var Spinner: any;  
  
@Component({  
  selector: 'cc-spinner',  
  templateUrl: './spinner.component.html',  
  styleUrls: ['./spinner.component.scss']  
})  
export class SpinnerComponent implements OnInit, OnDestroy {  
  
  private spinner: any;  
  show: boolean = false;  
  private element: any = null;  
  private subscription: Subscription = null;  
  
  @Input() lines: number = 12; // The number of lines to draw  
  @Input() length: number = 20; // The length of each line
```

```

@Input() width: number = 12; // The line thickness
@Input() radius: number = 50; // The radius of the inner circle
@Input() scale: number = 1.0; // Scales overall size of the spinner
@Input() corners: number = 1; // Corner roundness (0..1)
@Input() color: string = '#881635'; // #rgb or #rrggbb or array of colors
@Input() opacity: number = 0.25; // Opacity of the lines
@Input() rotate: number = 0; // The rotation offset
@Input() direction: number = 1; // 1: clockwise, -1: counterclockwise
@Input() speed: number = 0.8; // Rounds per second
@Input() trail: number = 60; // Afterglow percentage
@Input() fps: number = 20; // Frames per second when using setTimeout() as a fallback for CSS
@Input() className: string = 'spinner'; // The CSS class to assign to the spinner
@Input() top: string = '50%'; // Top position relative to parent
@Input() left: string = '50%'; // Left position relative to parent
@Input() shadow: boolean = true; // Whether to render a shadow
@Input() hwaccel: boolean = true; // Whether to use hardware acceleration
@Input() position: string = 'absolute'; // Element positioning

constructor(
  private spinnerElement: ElementRef,
  private spinnerService: SpinnerService
) {
  this.element = spinnerElement.nativeElement;
}

ngOnInit() {
  this.initSpinner();
  this.createServiceSubscription();
}

private initSpinner() {
  const options = {
    lines: this.lines,
    length: this.length,
    width: this.width,
    radius: this.radius,
    scale: this.scale,
    corners: this.corners,
    color: this.color,
    opacity: this.opacity,
    rotate: this.rotate,
    direction: this.direction,
    speed: this.speed,
    trail: this.trail,
    fps: this.fps,
    zIndex: 2e9, // Artificially high z-index to keep on top
    className: this.className,
    top: this.top,
    left: this.left,
    shadow: this.shadow,
    hwaccel: this.hwaccel,
    position: this.position
  };
}

```

```

        console.log('Creating spinner with options:');
        // console.log(JSON.stringify((options)));
        this.spinner = new Spinner(options);
    }

private createServiceSubscription() {
    this.subscription = this.spinnerService.spinnerObservable.subscribe(showSpinner => {
        if (showSpinner) {
            this.startSpinner();
        } else {
            this.stopSpinner();
        }
    });
}

ngOnDestroy() {
    this.subscription.unsubscribe();
}

startSpinner() {
    this.show = true;
    this.spinner.spin(this.element.firstChild);
}

stopSpinner() {
    this.show = false;
    this.spinner.stop();
}
}

```

Edit the **spinner.component.spec.ts** file.

Replace the entire contents with the following:

```

import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { SpinnerComponent } from './spinner.component';
import { SpinnerService } from '../../services/spinner.service';

describe('SpinnerComponent', () => {
    let component: SpinnerComponent;
    let fixture: ComponentFixture<SpinnerComponent>

    beforeEach(async(() => {
        TestBed.configureTestingModule({
            declarations: [
                SpinnerComponent
            ],
            providers: [
                SpinnerService
            ]
        })
    }));

```

```

.compileComponents();
});

beforeEach(() => {
  fixture = TestBed.createComponent(SpinnerComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});
});

```

Edit the **shared.module.ts** file.

Modify the code as indicated by the following:

```

1 import { NgModule, ModuleWithProviders } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 import { PageHeaderComponent } from './page-header/page-header.component';
5 import { PageFooterComponent } from './page-footer/page-footer.component';
6 +import { SpinnerComponent } from './spinner/spinner.component';
7
8 @NgModule({
9   imports: [
10     CommonModule,
11   ],
12   declarations: [
13     PageHeaderComponent,
14 +   PageFooterComponent,
15 +   SpinnerComponent
16   ],
17   exports: [
18     PageHeaderComponent,
19 +   PageFooterComponent,
20 +   SpinnerComponent
21   ]
22 })
23 export class SharedModule {
24   static forRoot(): ModuleWithProviders {
25     return {
26       ngModule: SharedModule,
27       providers: [
28
29     ]
30   };
31 }
32 }
33

```

Edit the **app.component.html** file.

Make the changes as indicated by the following:

```
1 <div class="spacer-15"></div>
2 +<cc-spinner [attr.radius]="40" [attr.lines]="12" [attr.width]="16" [attr.length]="40"
3 + [attr.opacity]="0.25" [attr.shadow]="true"></cc-spinner>
4 <div class="app-container">
5   <div class="app-inner-container">
6     <cc-page-header></cc-page-header>
7     <div class="spacer-15"></div>
8     <div class="page-container">
9       <router-outlet></router-outlet>
10    </div>
11    <div class="spacer-15"></div>
12    <cc-page-footer></cc-page-footer>
13  </div>
14 </div>
15 <div class="spacer-15"></div>
16
```

Edit the **app.component.ts** file.

Make the changes as indicated by the following:

```
1 import { Component, OnInit, OnDestroy } from '@angular/core';
2 import { Title } from '@angular/platform-browser';
3 import { Router, NavigationStart, NavigationEnd, ActivatedRouteSnapshot } from '@angular/router';
4 import { Observable } from 'rxjs/Observable';
5
6 +import { SpinnerService } from './services/spinner.service';
7 +
8 @Component({
9   selector: 'cc-root',
10  templateUrl: './app.component.html',
11  styleUrls: ['./app.component.scss']
12 })
13 export class AppComponent implements OnInit, OnDestroy {
14
15   title: Observable<string>;
16   titleSubscription: any;
17 + spinnerSubscription: any;
18
19   public constructor(
20     private router: Router,
21 +   private titleService: Title,
22 +   private spinnerService: SpinnerService
23   ) { }
24
25   ngOnInit() {
26
27     this.title = this.router.events
28       .filter((event) => event instanceof NavigationEnd)
29       .map(() => {
30         const routeSnapshot = this.router.routerState.snapshot.root;
31         return this.getDeepestTitle(routeSnapshot);
32       });
33
34     this.titleSubscription = this.title.subscribe((title) => {
35       const titleComplete = title && title.length > 0 ? 'Camp Comfy - ' + title : 'Camp Comfy';
36       this.titleService.setTitle(titleComplete);
37     });
38 +
39 +   this.handleRouterEvents();
40   }
41 }
```

```
42     ngOnDestroy() {
43         if (this.titleSubscription) {
44             this.titleSubscription.unsubscribe();
45         }
46 +        if (this.spinnerSubscription) {
47 +            this.spinnerSubscription.unsubscribe();
48     }
49 }
50
51     private getDeepestTitle(routeSnapshot: ActivatedRouteSnapshot) {
52         const title = routeSnapshot.firstChild && routeSnapshot.firstChild.data &&
53             routeSnapshot.firstChild.data.title ? routeSnapshot.firstChild.data.title : '';
54         return title;
55     }
56 +
57     handleRouterEvents() {
58 +
59         this.spinnerSubscription = this.router.events.subscribe((evt) => {
60             if (evt instanceof NavigationStart) {
61                 this.spinnerService.show();
62             }
63             if (evt instanceof NavigationEnd) {
64                 this.spinnerService.hide();
65                 if (window) {
66                     window.scrollTo(0, 0);
67                 }
68             }
69         },
70         (err) => {
71             console.log(err);
72         });
73     }
74 }
75
```

Edit the **app.component.spec.ts** file.

Make the changes indicated by the following:

```
1 import { TestBed, async } from '@angular/core/testing';
2 import { RouterTestingModule } from '@angular/router/testing';
3
4 import './rxjs-extensions';
5
6 import { SharedModule } from './shared/shared.module';
7
8 import { AppComponent } from './app.component';
9
10 +import { SpinnerService } from './services/spinner.service';
11 +
12 describe('AppComponent', () => {
13   beforeEach(async(() => {
14     TestBed.configureTestingModule({
15       imports: [
16         RouterTestingModule,
17         SharedModule
18       ],
19       declarations: [
20         AppComponent
21       ],
22 +       providers: [
23 +         SpinnerService
24 +       ]
25     }).compileComponents();
26   }));
27
28   it('should create the app', async(() => {
29     const fixture = TestBed.createComponent(AppComponent);
30     const app = fixture.debugElement.componentInstance;
31     expect(app).toBeTruthy();
32   }));
33 });
34
```

Edit the **search-flickr.component.ts** file.

Make the changes indicated by the following:

```
1 import { Component, OnInit, OnDestroy } from '@angular/core';
2 import { Router } from '@angular/router';
3 import { Observable } from 'rxjs/Observable';
4
5 import { FlickrService } from '../services/flickr.service';
6 +import { SpinnerService } from '../services/spinner.service';
7
8 @Component({
9   selector: 'cc-search-flickr',
10  templateUrl: './search-flickr.component.html',
11  styleUrls: ['./search-flickr.component.scss']
12 })
13 export class SearchFlickrComponent implements OnInit, OnDestroy {
14
15   searchString: string = '';
16   photos: any[] = [];
17
18   constructor(
19     private flickrService: FlickrService,
20 +   private router: Router,
21 +   private spinnerService: SpinnerService
22   ) { }
23
24   ngOnInit() {
25   }
26
27   ngOnDestroy() {
28   }
29
30   search() {
31
32     this.photos = [];
33
34     if (this.searchString.length === 0) {
35       alert('Please first enter a search query.');
36       return;
37     }
38   }

```

```
39
40 +     this.spinnerService.show();
41 +
42     this.flickrService.search(this.searchString).subscribe(
43         data => {
44             if (data.length === 0) {
45                 console.log('no photos matched your search query.');
46             }
47             else {
48                 console.log(data);
49                 this.photos = data;
50                 console.log(this.photos);
51             }
52         },
53         err => {
54             console.log(err);
55 +             this.spinnerService.hide();
56             this.router.navigate(['/error']);
57         },
58         () => {
59             console.log('Flickr search complete');
60 +             this.spinnerService.hide();
61         }
62     );
63 }
64 }
```

Edit the **search-flickr.component.spec.ts** file.

Add the spinner service to the providers list.

Generate Mock Data

Saturday, July 22, 2017 9:04 AM

Here are some online tools you can use to generate mock data in JSON file format:

<https://www.mockaroo.com/>

<http://www.json-generator.com/>

<http://www.theonegenerator.com/geradordejson>

<https://www.site24x7.com/tools/json-generator.html>

Setup A JSON Server

Sunday, August 6, 2017 8:19 AM

JSON Server is an npm package that enables you to create a fully functioning RESTful JSON web service using the JSON data file(s) you provide.

Located at GitHub:

<https://github.com/typicode/json-server>

You can install globally using the following:

`npm install -g json-server`

Use Fake Data Generator Libraries

Sunday, August 6, 2017 8:52 AM

You can use a JavaScript library to generate fake JSON data.

Here are several for you to explore:

Faker

<https://github.com/Marak/faker.js>

Casual

<https://github.com/boolearn/casual>

Chance

<https://github.com/chancejs/chancejs>

JSON Schema Faker

<https://github.com/json-schema-faker/json-schema-faker>

Use Freely Available Services

Sunday, August 6, 2017 8:20 AM

Search for free web service APIs.

Here are some examples:

Flickr (images): <https://www.flickr.com/services/api/>

SWAPI (The Star Wars API): <https://swapi.co/>

Google Maps: <https://developers.google.com/maps/>

YouTube: <https://developers.google.com/youtube/>

OpenWeatherMap: <https://openweathermap.org/pric>

Here's a large list of free web services:

<https://www.programmableweb.com/apis/directory>

Reduce The Size Of Your Images

Sunday, August 6, 2017 8:24 AM

Bitmap images, especially PNG images, are often much large than they need to be. You can reduce the total byte size of your images by running them through an optimizer process. Here are some image optimizers for you to explore:

nQuant

<https://nquant.codeplex.com/>

PNGCrush

<http://pngcrush.com/>

PNGGauntlet

<https://pnggauntlet.com/>

Sprites Reduce Total Site Downloads

Sunday, August 6, 2017 8:25 AM

Combining your icons into a single sprite file can significantly reduce the number of downloads when your site loads in a browser.

Most browsers only support 5 to 6 concurrent file downloads, and wait until all files have completely finished downloading before getting the next set of files to download.

Here are some sprite generators for you to explore:

<https://spritegenerator.codeplex.com/>

<http://spritegen.website-performance.org/>

Bring Along The High Quality, Lightweight Gear

Saturday, July 22, 2017 9:06 AM

SVG, or Scalable Vector Graphics, offers a way to do full resolution graphical elements, no matter what size screen, what zoom level, or what resolution your user's device has.

With SVG files you can actually style shapes with CSS, and make them interactive with JavaScript by attaching event handlers to SVG nodes.

SVG Files can be significantly smaller than an equivalent resolution bitmap image.

Ture Story: I once worked on a project where the initially produced PNG graphic was over 2MB in size. We refactored the image as an SVG graphic, and its size was reduced to just over 37KB. That was a 98% reduction in download size of the graphic file.

Minimize Your Site Resources

Sunday, August 6, 2017 8:29 AM

In addition to commenting out the unwanted Bootstrap styles from being compiled into your camp site stylesheet, you have other means available to you via the angular-cli command line tool to squeeze more unused stuff out of your final build file results.

To do a production build, you must add the `--prod` argument to the angular-cli command line call, like the following:

```
ng build --prod
```

The angular-cli tool ensures all builds make use of bundling and limited tree-shaking, while `--prod` builds also run limited dead code elimination via UglifyJS.

Resources in CSS, such as images and fonts, will be copied over automatically as part of a build. If a resource is less than 10kb it will also be in-lined.

Build Your Own Custom Font Sets

Sunday, August 6, 2017 8:30 AM

Icon fonts have many advantages with the biggest being their free scalability to any size, while at the same time reducing server requests to an absolute minimum. However, you will not need the whole font set, and you may need icons from different font sets. Font generators can help you with both of these challenges by producing custom fonts sets and CSS. Here are some examples:

IcoMoon

<https://icomoon.io/>

Fontello

<http://fontello.com/>

Fontastic

<http://fontastic.me/>

Generate Your FavIcon

Sunday, August 6, 2017 8:31 AM

Try using the following FavIcon Generator

<http://realfavicongenerator.net/>

Git Your Camp Site Deployed

Friday, July 21, 2017 2:03 PM

Edit the package.json file, and add the following:

```
5   "scripts": {  
6     "ngver": "ng -v",  
7     "start": "ng serve --open",  
8     "start:prod": "ng serve --prod --open",  
9     "hmr": "ng serve --hmr -e=hmr --open",  
10    "build": "ng build",  
11    "build:prod": "ng build --prod",  
12    "build:prod:folder": "ng build --prod --base-href=/my-app/",  
13 +   "git": "ng build --prod --output-path docs --base-href=./",  
14    "test": "ng test",  
15    "test:ci": "ng test --browsers=ChromeNoSandboxHeadless --watch=false",  
16    "lint": "ng lint --type-check --format stylish",  
17    "lint:fix": "ng lint --type-check --fix --format stylish",  
18    "lint:ci": "ng lint --type-check",  
19    "e2e": "ng e2e",  
20    "e2e:ci": "ng e2e --config=protractor-ci.conf.js"  
21  },
```

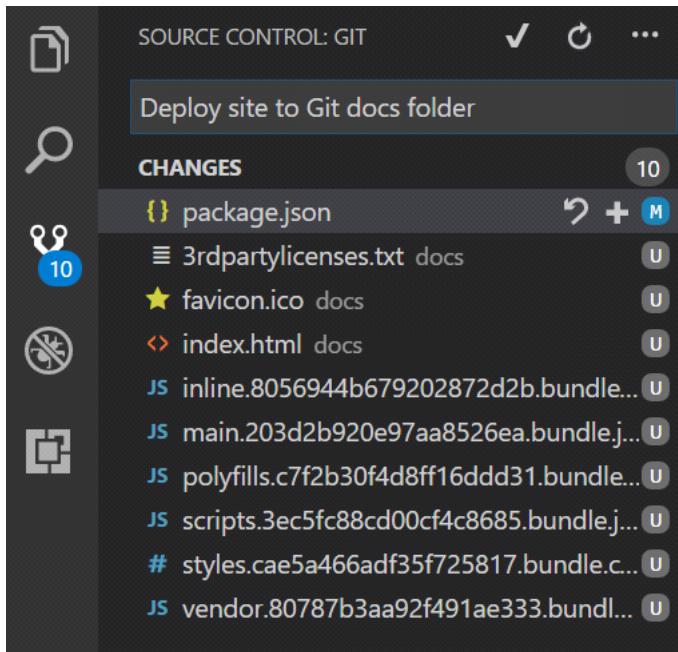
"git": "ng build --prod --output-path docs --base-href=./",

Run the Git script.

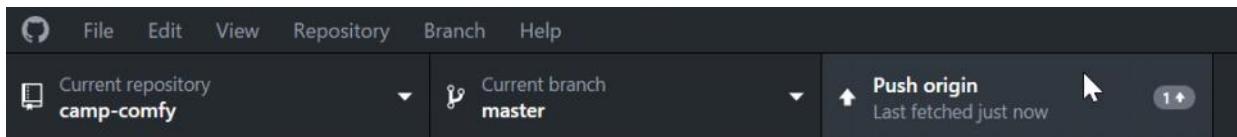
npm run git

```
C:\srcCC\camp-comfy>npm run git  
> camp-comfy@0.0.0 git C:\srcCC\camp-comfy  
> ng build --prod --output-path docs --base-href camp-comfy  
  
Hash: 51b9117099e130f85e47  
Time: 16586ms  
chunk  {0} polyfills.c7f2b30f4d8ff16ddd31.bundle.js (polyfills) 252 kB {5} [initial] [rendered]  
chunk  {1} main.203d2b920e97aa8526ea.bundle.js (main) 18.4 kB {4} [initial] [rendered]  
chunk  {2} scripts.3ec5fc88cd00cf4c8685.bundle.js (scripts) 146 kB {5} [initial] [rendered]  
chunk  {3} styles.cae5a466adf35f725817.bundle.css (styles) 69 bytes {5} [initial] [rendered]  
chunk  {4} vendor.80787b3aa92f491ae333.bundle.js (vendor) 1.15 MB [initial] [rendered]  
chunk  {5} inline.8056944b679202872d2b.bundle.js (inline) 0 bytes [entry] [rendered]
```

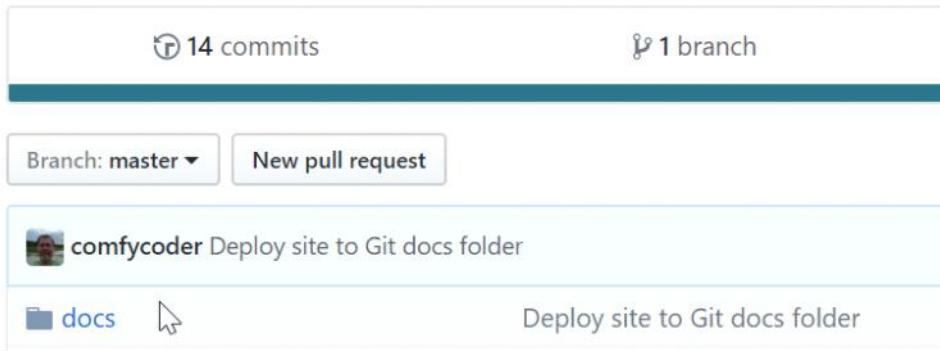
Stage and Commit your compile site code to Git local.



Publish your changes to your remote Git repository.



The doc folder should appear in the Git repository.



Open the Setting page.



In the GitHub Pages section, choose the option called:

master branch /docs folder

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more](#).

[None ▾](#)

[Save](#)

Select source

X

master branch

Use the master branch for GitHub Pages.

master branch /docs folder

Use only the /docs folder for GitHub Pages.



✓ None

Disable GitHub Pages.

D

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more](#).

[master branch /docs folder ▾](#)

[Save](#)



Theme Chooser

Select a theme to build your site with a Jekyll theme using the master branch /docs folder. [Learn more](#).

[Choose a theme](#)

Click the Save button.

Open a Browser and enter the path to the site using the following format for the url:

<https://comfycoder.github.io/camp-comfy/>

Change The Angular Application Prefix

Saturday, July 29, 2017 11:15 AM

You may wish to change the prefix that the angular-cli automatically assigns to your components when it creates them.

You should always try to make a decision for your prefix before you create a new angular application, as this will save you from having to make many changes later on.

You must make changes to various files, as follows:

Edit the file called .angular-cli.json, set the prefix value to the desired characters.

e.g., change from "app" to "cc":

```
20 |     "prefix": "cc",
```

Edit the file called tslint.json, search for the directive selector and component-selector section and set the desired characters.

e.g., change from "app" to "cc":

```
117 |     "directive-selector": [
118 |       true,
119 |       "attribute",
120 +     "cc",
121 |       "camelCase"
122 |     ],
123 |     "component-selector": [
124 |       true,
125 |       "element",
126 +     "cc",
127 |       "kebab-case"
128 |     ],
```

Edit the file called index.html, and change the component selector prefix to the desired characters.

e.g., change from "app" to "cc":

```
19 + <cc-root></cc-root>
```

Finally, you must search through all of you code, html, test, and e2e files and change the component selector prefix to the desired characters.

e.g., change from "app" to "cc":

```
3 @Component({
4 + selector: 'cc-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
```

Monitor Camp Site Usage And Report Issues

Saturday, July 22, 2017 9:06 AM

Track.js

<https://trackjs.com/>

Helps you track and resolve client-side errors.

Application Insights for web pages

Find out about the performance and usage of your web page or app, get timings of page loads and AJAX calls, counts and details of browser exceptions and AJAX failures, as well as users and session counts.

<https://docs.microsoft.com/en-us/azure/application-insights/app-insights-javascript>

A Simple ASP.NET Core Site To Host Your Angular SPA

Sunday, August 6, 2017 8:35 AM

See example in my GitHub repository: Pending

Learning Site Links

Sunday, August 6, 2017 9:44 AM

Angular.io

<https://cli.angular.io/> (Angular CLI)
<https://github.com/angular/angular-cli/wiki> (Angular CLI)
<https://angular.io/guide/quickstart>
<https://angular.io/tutorial>

Rangle.io Angular Training Book

<https://angular-2-training-book.rangle.io/>

Pluralsight Videos

Sunday, August 6, 2017 9:45 AM

Pluralsight offers you the best training you will find for your money. The training comes in short video segments organized by category. You can watch them over and over as many times as needed to reinforce your knowledge and get just in time architectural guidance from the best.

Here are my favorite videos from my Angular journey:

Angular CLI - John Papa (do every task in this video to quickly master the use of angular-cli)

<https://app.pluralsight.com/library/courses/angular-cli/table-of-contents>

Angular: Getting Started - Deborah Kurata

<https://app.pluralsight.com/library/courses/angular-2-getting-started-update/table-of-contents>

Angular Fundamentals - Joe Eames

<https://app.pluralsight.com/library/courses/angular-fundamentals/table-of-contents>

Angular Routing - Deborah Kurata

<https://app.pluralsight.com/library/courses/angular-routing/table-of-contents>

Play by Play: Angular 2/RxJS/HTTP and RESTful Services - John Papa and Dan Wahlin

<https://app.pluralsight.com/library/courses/play-by-play-angular-2-rxjs-http-restful-services-john-papa-dan-wahlin/table-of-contents>

Angular Reactive Forms - Deborah Kurata

<https://app.pluralsight.com/library/courses/angular-2-reactive-forms/table-of-contents>

Shiny New Tools

Sunday, August 6, 2017 9:45 AM

Online VS Code IDE for Angular & React

<https://stackblitz.com/>

Custom Components

Thursday, October 19, 2017 7:47 AM

<http://blog.rangle.io/angular-2-ngmodel-and-custom-form-components/>

<https://github.com/okunishinishi/node-stringcase>