The Angular CLI

ISAAC MARTINEZ
Software Architect
TravelTab

Twitter: @IkeMtz
Email: pcalchi@aim.com

Learning JavaScript in 2017

Glossary of terms

NPM – A package manager similar to NuGet or Bower

https://www.npmjs.com/

TypeScript – A language that provides compile time safety checks and transpiles code to JavaScript
https://www.typescriptlang.org/

Node.Js – Server and client JavaScript environment similar to .net Framework.
https://nodejs.org/en/

Installing the angular cli

```
npm install -g @angular/cli@latest
```

Requires Node.Js to be installed
https://nodejs.org/en/

Currently the CLI is at RC1 as of 3/3/17

Creating your first CLI project

Angular2:

```
ng new {app name} --style scss
```

Angular4:

```
ng new {app name} --style scss --ng4 true
```

Scaffolds Angular 2/4 project

Sets up project to use Scss (Less/Sass and plain CSS are also options)

Sets up project for unit and e2e (End To End) testing

Sets up a development server using Node.Js and BrowserSync

Things to consider

Angular CLI sets up your package.json file to use semver (semantic versioning); this has the potential of breaking your project if an underlying dependency breaks. Consider shrink wrapping.

```
npm shrinkwrap
```

There are other versions of the CLI available (with some differences)

- NativeScript CLI – Tooling for building NativeScript Angular applications that run on mobile platforms.
- Universal CLI – Tooling for building Universal Angular applications.

Starting Up the test environment

```
ng serve
```

Branch: Step1

Adding 3rd Party Libraries

Branch: Step2

Edit the Angular-cli.json file and add the relevant assets to the respective styles and scripts arrays.

We’ll be adding Bootstrap 4 currently in alpha 6 to our project.

```
npm install –save-dev bootstrap@4.0.0-alpha.6
```

SCAFFOLDING

```
ng g {type} {name}
```

Module

Component

Route

Service

Directive

Pipe

Note: In most cases the CLI will generate unit tests for the generated code.

The Hero-Form

```
ng g class hero
ng g component hero-form
```

Branch: Step4

STEP 4: CLI Linting

Branch: Step4a – Lint FAIL
Branch: Step4b – Lint Pass

```
ng lint
```

NG Lint will run TS Lint on your code and report back code that doesn't follow your rules. Rules are specified in tslint.json and are customizable.

Strongly recommend TSLint VS Code extension.
https://marketplace.visualstudio.com/items?itemName=eg2.tslint

STEP 5: CLI Unit Testing

Branch: Step5

```
ng test
```

NG Test will execute a Karma test run on your code using your spec files. These spec files are generated by the CLI scaffolding.

99.9% of the time you will have to edit these files in order for your test to pass.

Supports “--watch”, save your files and your test will automatically rerun.

Supports “--code-coverage”, will generate html report showing code coverage percentage and which lines are not covered.

Supports multiple browsers

STEP 6: CLI E2E Testing

Branch: Step6

```
ng e2e
```

Currently there is a bug in the app.e2e-spec.ts file. Repository has fix.

Runs Protractor and allows you to create fully automated UI tests across multiple browsers.

Protractor documentation is seriously out of date and a few of the features don’t work on Angular 2 and above.

There are many more features!!!

- Environment specific builds
- AOT (Ahead of Time) compilation
- Hot Module Reloading (requires some WebPack2 customization)
- Support for Internalization
- Less, Sass, Scss pre-processor compilation

References

Angular 2 Docs https://angular.io/docs/ts/latest/

Angular CLI https://github.com/angular/angular-cli

Angular 2 Tour of Heroes https://angular.io/docs/ts/latest/tutorial/

TypeScript Tutorial https://www.typescriptlang.org/docs/tutorial.html

Less Tutorial http://lesscss.org