

BUILDING COMPONENT LIBRARIES

WITH THE ANGULAR CLI



BRENDEN NIEDERMEYER

Expeditors®

CARGOSIGNAL™



Angular Seattle July 2018 | Brenden Niedermeyer

OVERVIEW

1. What's new with the Angular CLI
2. Generating a library
3. Using your library's components
4. Using your library in another project

WHAT'S NEW WITH THE ANGULAR CLI?

Angular CLI v6.0 was released on May 3

SCHEMATICS



NG ADD



NG UPDATE



```
brendenniedermeyer@Brendens-MacBook-Pro: ~/Desktop/demo-app (zsh)
~/Desktop/demo-app ➜ master ➜
```

NG GENERATE LIBRARY

A new command that adds a library to CLI projects
ng-packagr is used to build the library

```
$ ng generate library <library-name>
```

Creates a preconfigured library project in projects/<library-name>

OPEN EDITORS

counter-button.component.ts projects/... U

DEMO

dist e2e node_modules projects my-lib src lib counter-button counter-button.component.c... U counter-button.component.h... U counter-button.component.s... U A counter-button.component.ts U my-lib.module.ts TS public_api.ts TS test.ts K karma.conf.js {...} ng-package.json {...} ng-package.prod.json JS package.json {...} tsconfig.lib.json {...} tsconfig.spec.json {...} tslint.json

src

1 import { Component, OnInit, EventEmitter, Output } from '@angular/core';
2
3 @Component({
4 selector: 'my-lib-counter-button',
5 templateUrl: './counter-button.component.html',
6 styleUrls: ['./counter-button.component.css']
7 })
8 export class CounterButtonComponent implements OnInit {
9 @Output() countChanged: EventEmitter<number> = new EventEmitter();
10 clickCount = 0;
11
12 constructor() {}
13
14 ngOnInit() {}
15
16 /
17 * Increment the click count when the button is clicked and emits an event
18 * notifying the parent component of the new count value
19 */
20 handleClick() {
21 this.clickCount++;
22 this.countChanged.emit(this.clickCount);
23 }
24
25 }
26
27 }

DEMO

BUILDING YOUR COMPONENT LIBRARY

```
$ ng build <library-name>
```

This builds the library in dist/projects/<library-name>

TWO WAYS TO USE YOUR LIBRARY

Angular Workspace:

- ▶ built library is included in `.tsconfig.json`
- ▶ must be built before the project can be built

```
import {LibraryModule} from '<library-name>';

@NgModule({
  imports: [LibraryModule],
  ...
})
```

TWO WAYS TO USE YOUR LIBRARY

NPM

- ▶ build library with `ng build <library-name>`
- ▶ `npm publish`

TWO WAYS TO USE YOUR LIBRARY

LOCAL

Use `npm link` to use in other projects, or the generated local project to test

```
$ cd dist/<library-name>
$ npm link
$ cd <new app dir>
$ npm link <library-name>
```

GOTCHAS



GOTCHAS

Bundling Assets

```
{  
  "bundle": "cp lib_assets dist/my-lib"  
}
```

scss-bundle

GOTCHAS

Dependencies

- ▶ Use root package.json to manage dev dependencies
- ▶ Keep peer dependencies in your library up to date with dependencies in root package.json

GOTCHAS

Dependencies

- ▶ If declaring a dependency use `whitelistedNonPeerDependencies` in `ng-package.json`

ng-package.json

```
{  
  "$schema": "./node_modules/ng-packagr/package.schema.json",  
  "ngPackage": {  
    "whitelistedNonPeerDependencies": ["moment"]  
  }  
}
```

RESOURCES

- ▶ Official Angular CLI Documentation: <https://github.com/angular/angular-cli/wiki/stories-create-library>
- ▶ Code Samples/Slides: <https://github.com/bniedermeyer/angular-seattle-library-demo>
- ▶ Article: <http://bit.ly/angular-cli-libs>

