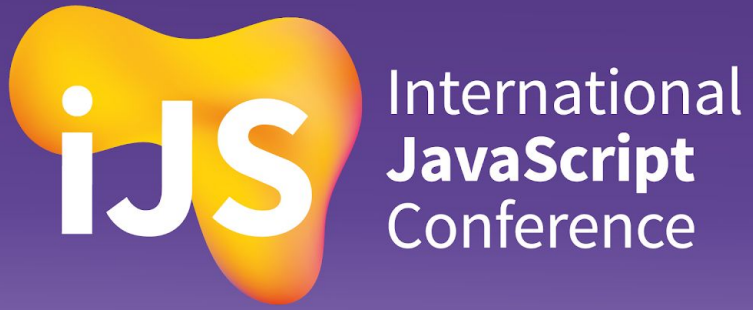


Angular Custom Libraries

Matt Vaughn
AngularArchitecture.com
@AngularArch
@Angularlicious

About me...

- Owned by Lukka the Husky
- Live and work in Colorado
- Developer since 1998
- Love tacos…
- Fly Fishing
- Saxophone...Smooth Jazz

What is a Library?

An Angular library is just a collection of things that can be consumed by multiple projects.

Angular CLI

A magical tool to create things Angular...

Why are they so useful?

Sharing and reusing code - many benefits.

Do you know...?

What does the `@angular/cli` give us?

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

```
user@buildmotion-pro ~ % ng --version

     _                      _                 ____ _     ___
    / \   _ __   __ _ _   _| | __ _ _ __     / ___| |   |_ _|
   / △ \ | '_ \ / _` | | | | |/ _` | '__|   | |   | |    | |
  / ___ \| | | | (_| | |_| | | (_| | |      | |___| |___ | |
 /_/   \_\_| |_|\__, |\__,_|_|\__,_|_|       \____|_____|___|
                |___/


Angular CLI: 11.2.14
Node: 16.9.1
OS: darwin x64

Angular:
...
Ivy Workspace:

Package                      Version
-------------------------------------------------------
@angular-devkit/architect    0.1102.14 (cli-only)
@angular-devkit/core         11.2.14 (cli-only)
@angular-devkit/schematics   11.2.14 (cli-only)
@schematics/angular          11.2.14 (cli-only)
@schematics/update           0.1102.14 (cli-only)

user@buildmotion-pro ~ % ng new some-name-here
```

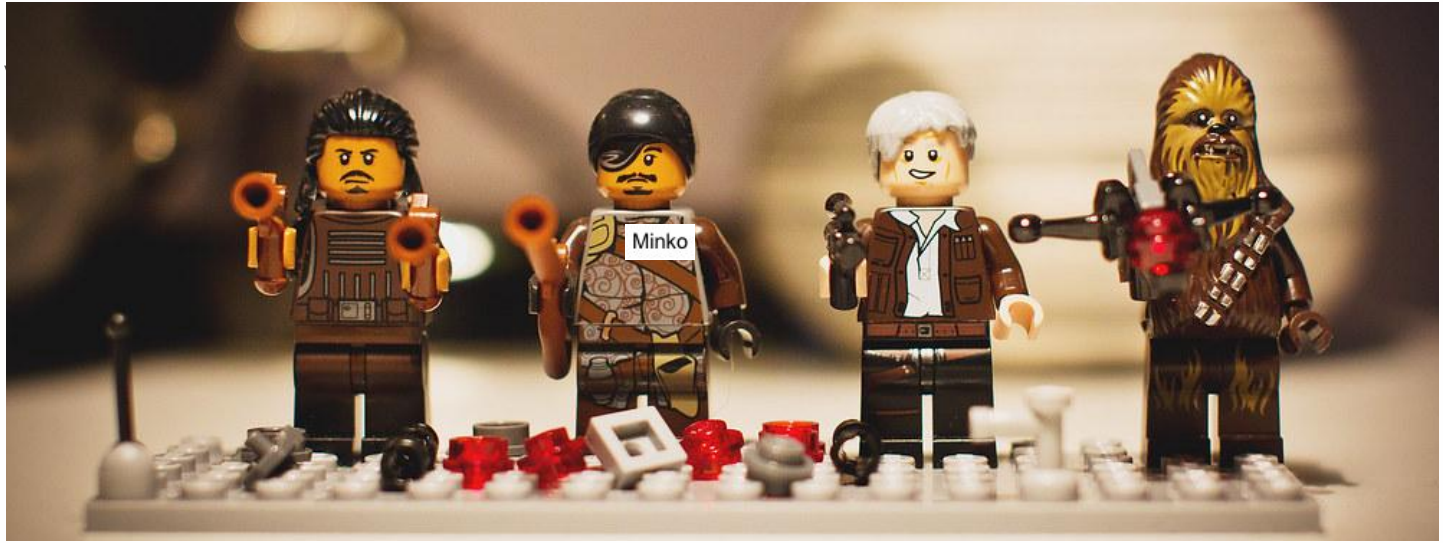
Do you know...?

What does the `ng new` CLI command give us?

```
ng new some-name-here
```

Angular Version 6 gave us the ***Workspace***?

May the 4th (2018) be with you.



{} angular.json > ...

Matt Vaughn, 3 hours ago | 2 authors (Matt Vaughn and others)

```
1   {      matt, 2 weeks ago • add ng ws default
2     "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
3     "version": 1,
4     "newProjectRoot": "projects",
5     "projects": {
6       "errorHandler": {
7         "projectType": "library",
8         "root": "projects/error-handler",
9         "sourceRoot": "projects/error-handler/src",
10        "prefix": "lib",
11  >     "architect": {…
44        }
45      },
46      "portal": {
47        "projectType": "application",
48  >     "schematics": {…
52        },
53        "root": "projects/portal",
54        "sourceRoot": "projects/portal/src",
55        "prefix": "app",
56  >     "architect": {…
164       }
165     }
166   },
167   "defaultProject": "errorHandler"
168 }
```

What do we get with an Angular Workspace?

- Multiple projects within the same development environment.
  - Monorepo
- New project type: `library`
- Updated folder structure (code organization)

user — -zsh — 141×39

```
user@buildmotion-pro ~ % ng new some-name-here -d
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? SCSS   [ https://sass-lang.com/documentation/syntax
      ]
CREATE some-name-here/README.md (1022 bytes)
CREATE some-name-here/.editorconfig (274 bytes)
CREATE some-name-here/.gitignore (631 bytes)
CREATE some-name-here/angular.json (3769 bytes)
CREATE some-name-here/package.json (1215 bytes)
CREATE some-name-here/tsconfig.json (783 bytes)
CREATE some-name-here/tslint.json (3185 bytes)
CREATE some-name-here/.browserslistrc (703 bytes)
CREATE some-name-here/karma.conf.js (1431 bytes)
CREATE some-name-here/tsconfig.app.json (287 bytes)
CREATE some-name-here/tsconfig.spec.json (333 bytes)
CREATE some-name-here/src/favicon.ico (948 bytes)
CREATE some-name-here/src/index.html (298 bytes)
CREATE some-name-here/src/main.ts (372 bytes)
CREATE some-name-here/src/polyfills.ts (2830 bytes)
CREATE some-name-here/src/styles.scss (80 bytes)
CREATE some-name-here/src/test.ts (753 bytes)
CREATE some-name-here/src/assets/.gitkeep (0 bytes)
CREATE some-name-here/src/environments/environment.prod.ts (51 bytes)
CREATE some-name-here/src/environments/environment.ts (662 bytes)
CREATE some-name-here/src/app/app-routing.module.ts (245 bytes)
CREATE some-name-here/src/app/app.module.ts (393 bytes)
CREATE some-name-here/src/app/app.component.scss (0 bytes)
CREATE some-name-here/src/app/app.component.html (24955 bytes)
CREATE some-name-here/src/app/app.component.spec.ts (1081 bytes)
CREATE some-name-here/src/app/app.component.ts (219 bytes)
CREATE some-name-here/e2e/protractor.conf.js (904 bytes)
CREATE some-name-here/e2e/tsconfig.json (274 bytes)
CREATE some-name-here/e2e/src/app.e2e-spec.ts (665 bytes)
CREATE some-name-here/e2e/src/app.po.ts (274 bytes)

NOTE: The "dryRun" flag means no changes were made.
```

angular.json

src

```
user@buildmotion-pro ~ % ng new some-name-here --create-application=false -d
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
CREATE some-name-here/README.md (1022 bytes)
CREATE some-name-here/.editorconfig (274 bytes)
CREATE some-name-here/.gitignore (631 bytes)
CREATE some-name-here/angular.json (139 bytes)
CREATE some-name-here/package.json (1143 bytes)
CREATE some-name-here/tsconfig.json (783 bytes)
CREATE some-name-here/tslint.json (2992 bytes)

NOTE: The "dryRun" flag means no changes were made.
```

angular.json

--dry-run

[ng new] Recap

Creates a Workspace for multi-project environment. New [library] project type.

Angular CLI

How to create new library projects....

Do you know...?

What does the `ng g library` CLI command give us?

```
ng generate library some-library-name --dry-run
```

New library project output:

```
user@buildmotion-pro angular-workspace % ng generate library some-library-name --dry-run
CREATE projects/some-library-name/README.md (1066 bytes)
CREATE projects/some-library-name/karma.conf.js (1438 bytes)
CREATE projects/some-library-name/ng-package.json (166 bytes)
CREATE projects/some-library-name/package.json (198 bytes)
CREATE projects/some-library-name/tsconfig.lib.json (540 bytes)
CREATE projects/some-library-name/tsconfig.lib.prod.json (230 bytes)
CREATE projects/some-library-name/tsconfig.spec.json (309 bytes)
CREATE projects/some-library-name/tslint.json (247 bytes)
CREATE projects/some-library-name/src/public-api.ts (199 bytes)
CREATE projects/some-library-name/src/test.ts (781 bytes)
CREATE projects/some-library-name/src/lib/some-library-name.module.ts (289 bytes)
CREATE projects/some-library-name/src/lib/some-library-name.component.spec.ts (691 bytes)
CREATE projects/some-library-name/src/lib/some-library-name.component.ts (296 bytes)
CREATE projects/some-library-name/src/lib/some-library-name.service.spec.ts (404 bytes)
CREATE projects/some-library-name/src/lib/some-library-name.service.ts (144 bytes)
UPDATE angular.json (6617 bytes)
UPDATE tsconfig.json (1034 bytes)

NOTE: The "dryRun" flag means no changes were made.
```

CLI

projects folder

updates files

tsconfig.JSON

tsconfig.json > { } compilerOptions

You, 2 minutes ago | 2 authors (matt and others)

```
/* To learn more about this file see: https://angular.
{
  "compileOnSave": false,
  "compilerOptions": {
    "baseUrl": "./",
    "outDir": "./dist/out-tsc",
    "forceConsistentCasingInFileNames": true,
    "paths": {
      "error-handler": [
        "dist/error-handler/error-handler",
        "dist/error-handler"
      ],
      "some-library-name": [
        "dist/some-library-name/some-library-name",
        "dist/some-library-name"
      ]
    },
```

Imports mapper

to

from

locations

Building Libraries

Make some use out of the `dist` folder.

Do you know...?

What does the `ng build errorHandler` CLI command give us?

The `ng-packagr` applies the Angular Package Format to the library.

```
"projects": {
  "errorHandler": {
    "projectType": "library",
    "root": "projects/error-handler",
    "sourceRoot": "projects/error-handler/src",
    "prefix": "lib",
    "architect": {
      ng run errorHandler:build
      "build": {
        "builder": "@angular-devkit/build-angular:ng-packagr",
        "options": {
          "tsConfig": "projects/error-handler/tsconfig.lib.json",
          "project": "projects/error-handler/ng-package.json"
        },
        "configurations": {
          ng run errorHandler:build:production
          "production": {…
          }
        }
      },
      ng run errorHandler:test
      "test": {…
      },
```

non-Angular package by David Herges

Angular Package Format

```
user@buildmotion-pro angular-workspace % ng build errorHandler
Building Angular Package
************************************************************************************
It is not recommended to publish Ivy libraries to NPM repositories.
Read more here: https://v9.angular.io/guide/ivy#maintaining-library-compatibility
************************************************************************************

------------------------------------------------------------------------------
Building entry point 'error-handler'
------------------------------------------------------------------------------
✔ Compiling TypeScript sources through NGC
⠹ Bundling to FESM2015(node:51338) [DEP0148] DeprecationWarning: Use of deprecated folder mapping "./" i
odule resolution of the package at /Users/user/work/github/ijs-custom-angular-libraries/angular-workspac
ckage.json.
Update this package.json to use a subpath pattern like "./*".
(Use `node --trace-deprecation ...` to show where the warning was created)
✔ Bundling to FESM2015
✔ Bundling to UMD
✔ Minifying UMD bundle
✔ Writing package metadata
i Built error-handler

------------------------------------------------------------------------------
Built Angular Package
 - from: /Users/user/work/github/ijs-custom-angular-libraries/angular-workspace/projects/error-handler
 - to:   /Users/user/work/github/ijs-custom-angular-libraries/angular-workspace/dist/error-handler
------------------------------------------------------------------------------
```

CLI command

Angular compiler

Angular Package Format

Package Output

Do you know...?

Where does the `ng build` `errorHandler` CLI command put our package?

Package output in the [dist] folder.

```
.
├── error-handler
│   ├── bundles
│   │   ├── error-handler.umd.js
│   │   ├── error-handler.umd.js.map
│   │   ├── error-handler.umd.min.js
│   │   └── error-handler.umd.min.js.map
│   ├── esm2015
│   │   ├── lib
│   │   │   ├── error-handler.component.js
│   │   │   ├── error-handler.module.js
│   │   │   └── error-handler.service.js
│   │   ├── error-handler.js
│   │   └── public-api.js
│   ├── fesm2015
│   │   ├── error-handler.js
│   │   └── error-handler.js.map
│   ├── lib
│   │   ├── error-handler.component.d.ts
│   │   ├── error-handler.component.d.ts.map
│   │   ├── error-handler.module.d.ts
│   │   ├── error-handler.module.d.ts.map
│   │   ├── error-handler.service.d.ts
│   │   └── error-handler.service.d.ts.map
│   ├── error-handler.d.ts
│   ├── error-handler.d.ts.map
│   ├── package.json
│   ├── public-api.d.ts
│   ├── public-api.d.ts.map
│   └── README.md
```

The library output [dist] package.json file:

dist > error-handler > {} package.json > ...

```
1  {
2    "name": "error-handler",
3    "version": "0.0.1",
4    "peerDependencies": {
5      "@angular/common": "^11.2.14",
6      "@angular/core": "^11.2.14"
7    },
8    "dependencies": {
9      "tslib": "^2.0.0"
10   },
11   "main": "bundles/error-handler.umd.js",
12   "module": "fesm2015/error-handler.js",
13   "es2015": "fesm2015/error-handler.js",
14   "esm2015": "esm2015/error-handler.js",
15   "fesm2015": "fesm2015/error-handler.js",
16   "typings": "error-handler.d.ts",
17   "sideEffects": false,
     ▷ Debug
18   "scripts": {
19     "prepublishOnly": "node --eval \"console.error('ERROR: Trying to publish a package that
20   }
21 }
```

Require manual install

Indicates dependencies that will be installed automatically.

Entry points for different module loaders for the package.

The `public-api` exports the public members of the library.

dist > error-handler > TS public-api.d.ts

```
1  export * from './lib/error-handler.service';
2  export * from './lib/error-handler.component';
3  export * from './lib/error-handler.module';
4  //# sourceMappingURL=public-api.d.ts.map
```

CLI command recap...?

The CLI simplifies our workflow, right?

```
ng new angular-workspace --create-application-false
ng generate library error-handler
ng build errorHandler
```

Use the Library

How to use a new library project....

ErrorHandlerService in the Library

```
You, seconds ago | 3 authors (Matt Vaughn and others)
import { ErrorHandler, Injectable } from '@angular/core';

You, seconds ago | 3 authors (Matt Vaughn and others)
@Injectable({
  providedIn: 'root'
})
export class ErrorHandlerService implements ErrorHandler {

  constructor(      Matt Vaughn, 6 days ago • identify candidates
    // FIXME: consider injecting a logging service here; AN OPP FOR ANOTHER LIBRARY PROJECT! :)
  ) { }

  handleError(error: any): void {
    console.error(`ErrorHandlerService: ${error}`); // really? you'll want to do much more than this, ri
    // FIXME: USE A LOGGING SERVICE TO SEND THE INFORMATION HOME...KNOW AND UNDERSTAND THE HEALTH OF THE
  }
}
```

Do you know...?

How do we use the library in our application?

Import the service using the library name (tsconfig|paths)

projects > portal > src > app > TS app.module.ts > ...

```
1   import { ErrorHandler, NgModule } from '@angular/core';
2   import { BrowserModule } from '@angular/platform-browser';
3
4   import { AppRoutingModule } from './app-routing.module';
5   import { AppComponent } from './app.component';
6
7   import { ErrorHandlerService } from 'error-handler'        Matt Vaughn
8
    Matt Vaughn, 7 hours ago | 1 author (Matt Vaughn)
9   @NgModule({
10    declarations: [
11      AppComponent
12    ],
13    imports: [
14      BrowserModule,
15      AppRoutingModule
16    ],
17    providers: [
18      {
19        provide: ErrorHandler,
20        useClass: ErrorHandlerService
21      }
22    ],
23    bootstrap: [AppComponent]
24  })
25  export class AppModule { }
26
```

Library Demo

Inject the Library Service into the app...

localhost:4200

Welcome

portal app is running!

Resources

Here are some links to help you get started:

- Cause Trouble >
- Learn Angular >
- CLI Documentation >
- Angular Blog >

Next Steps

What do you want to do next with your app?

Elements | Console | Sources | Network | Performance | Memory | Application | Security

Page | Filesystem

- top
  - localhost:4200
    - (index)
    - main.js
    - polyfills.js
    - runtime.js
    - vendor.js
    - styles.css
  - webpack://

error-handler.service.ts × | app.component.ts | core.js

```
1  import { ErrorHandler, Injectable } from '@angular/core';
2
3  @Injectable({
4    providedIn: 'root'
5  })
6  export class ErrorHandlerService implements ErrorHandler {
7
8    constructor(
9      // FIXME: consider injecting a logging service here; AN OH
10   ) {
11     console.log(`Loading [ErrorHandlerService] during app ini
12   }
13
14   handleError(error: any): void {
15     console.error(`ErrorHandlerService: ${error}`); // really
16     // FIXME: USE A LOGGING SERVICE TO SEND THE INFORMATION H
17   }
18 }
19
```

{} Line 15, Column 5 (source mapped from main.js) Coverage: n/a

Console | Search | Issues

top ▾ | Filter | Default le

```
Loading [ErrorHandlerService] during app initialization.
Angular is running in development mode. Call enableProdMode() to enable production mode.
[WDS] Live Reloading enabled.
▼ErrorHandlerService: Error: Hello error...
  handleError                                  @ error-handler.service.ts:15
  handleError                                  @ core.js:10953
  executeListenerWithErrorHandling             @ core.js:15282
  wrapListenerIn_markDirtyAndPreventDefault    @ core.js:15317
  (anonymous)                                  @ platform-browser.js:592
```

Angular Library Types

Different flavors...kind of like ice-cream

Different libraries for different reasons...

- UI Components
- Cross-Cutting Concerns
- UI Feature(s)
- Domain Service
- Frameworks
- Validators

Identify Library Candidates

Be on the lookout for potential library candidates.

Do you see any potential candidates...

- **Code with similarities scattered throughout the codebase**
  - **Refactor to a single location**
  - **Isolate the code - single-source of truth**
  - **Determine the variations and refactor**
- **Multiple applications have similar needs (e.g., cross-cutting concerns)**
  - **Logging, security, error-handling, configuration, etc.**
- **Shared UI feature across different applications**

Publishing a Library

Sharing is caring.

Did you know...?

Need to modify `angularCompilerOptions` in the library's `tsconfig` file?

```
"compilationMode": "partial"
```

Publishing to NPM

1. create an account https://npmjs.com
2. create and build your library (i.e., package)
3. go to the dist folder and login: `npm login`
4. use the `npm publish` command to publish the package

@angular/core: package.json

- version
- description
- author
- license
- repository

```
{
  "name": "@angular/core",
  "version": "0.0.0-PLACEHOLDER",
  "description": "Angular - the core framework",
  "author": "angular",
  "license": "MIT",
  "engines": {
    "node": "^12.14.1 || >=14.0.0"
  },
  "dependencies": {
    "tslib": "^2.3.0"
  },
  "peerDependencies": {
    "rxjs": "^6.5.3 || ^7.0.0",
    "zone.js": "~0.11.4"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/angular/angular.git",
    "directory": "packages/core"
  },
  "ng-update": {
```

Library Versioning

Semantic version...

Did you know...?

You can use npm to version your package:

- `npm version patch`
- `npm version minor`
- `npm version major`

Code Organization Strategies

With Libraries....

Opportunities for code organization...

- **Monorepo: many projects, one repository**
  - **Developer workflow (efficient and effective)**
  - **No publish libraries**
- **Single implementations of common code**
  - **Well-tested, verified, higher quality**
- **Layered Architecture**
  - **UI/UX feature libraries**
  - **Domain libraries**

Angular Library Resources

More information....

More about Angular libraries...?

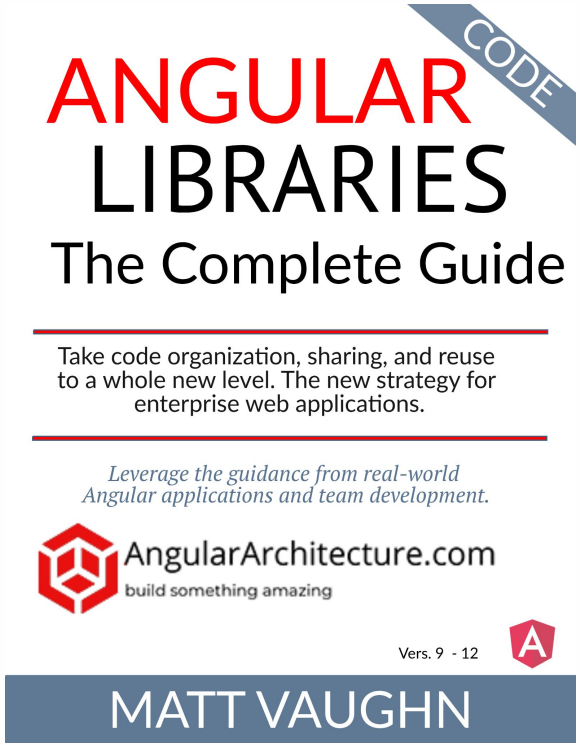
- Contributing packages to the registry - npmjs.com
- The Best Way To Architect Your Angular Libraries by Tomas Trajan
- Publish to NPM - npmjs.com
- Nrwl.io provides the Nx Workspace at Nx.dev
  - Advanced Workspace for Angular application and library projects
  - NestJS (backend)

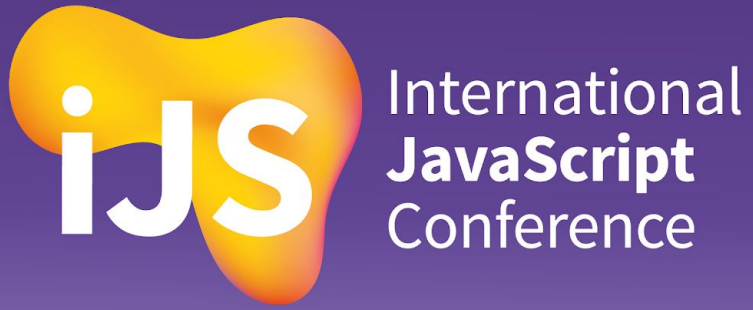
More Angular Library

Information....

Free Guide with more details about Angular Libs...target publish date in November 2021.

bit.ly/ijs-angular





Angular Custom Libraries

Matt Vaughn
AngularArchitecture.com
@AngularArch
@Angularlicious