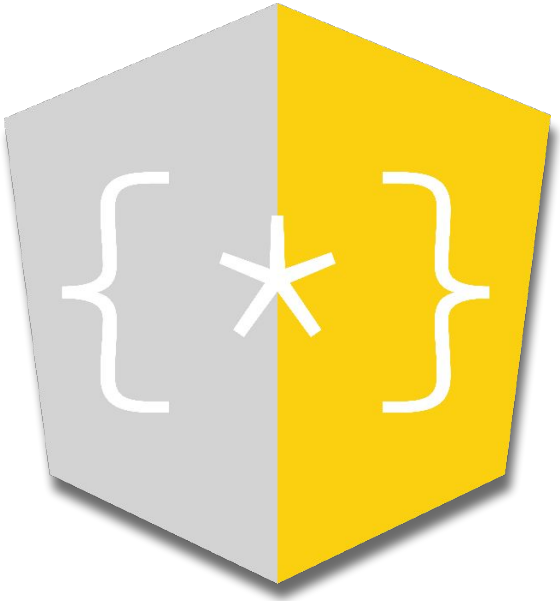


Custom Libraries

for Angular Applications

Matt Vaughn
@angularlicious
matt@angularlicio.us

hi!

I am Matt Vaughn

Developer, Speaker, Consultant, PodCaster, Musician, Owned by Lukka

 @angularlicious

 github.com/buildmotion

 http://www.angularlicio.us **OR** www.angularlicious.com

Lukka THE Husky

Profile:

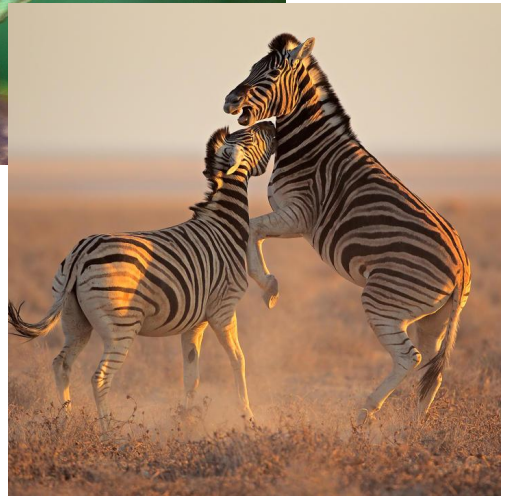
- 4 years old
- 75 pounds
- Instagram: **lukka_the_husky**


In a galaxy far far away...

Planet EARTH

Think about what we know about the planet Earth..

- Air and Weather
- Water
- Elements
- Animals and Insects



Name and Categorize

It's what we do.

Software DEVELOPMENT

The process of taking ideas or concepts to a working application - involving actions to ***design***, ***program***, test, and deliver.

Name and ***categorize*** elements in our code to represent ***things*** or to establish ***relationships*** between things.

?

WHAT?

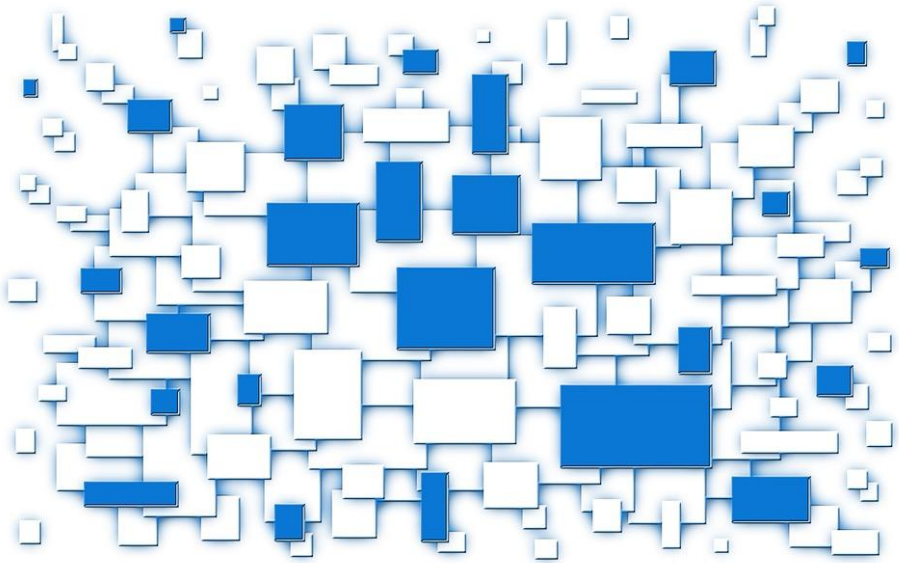
MODULES VS. LIBRARIES?

WHAT is A MODULE?

Definition: A module is a collection of ***related*** things - organized together to provide or perform some related functionalities.

Javascript: It is part of the ECMAScript specification 2015.

- ES6
- ESM6
- EcmaScript 2015



What is a LIBRARY?

A LIBRARY is a collection of useful things collected and ***distributed*** together - is ***consumable*** (think npm).

WHY?

MODULES, PACKAGES, LIBRARIES?

How **MANY** PACKAGES?

598,876

Packages in NPM as of 3/14/2018

558 Average Growth

Per day

107,313/96

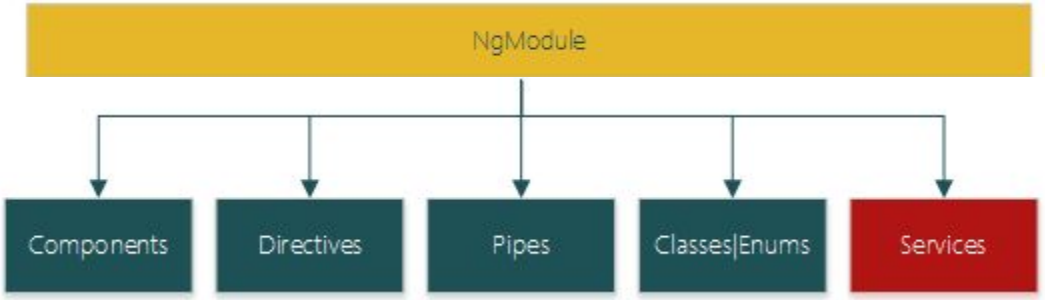
Compared to nuget (.NET)

@NgModule

TYPES

Not all modules are the same...hmmm?

@NgModule Members



Different TYPES of LIBS

Core/Shared

Core: App-specific items that are not 3rd-party or Angular.

Shared: Import Angular and other 3rd-party modules.

Feature

Modules that are specific to features of the application. Domain specific modules.

- Domain Service
- UI

Foundational

Modules that contain items that provide infrastructure, base or common behavior for Angular or application elements (i.e., components, services, actions, or HTTP services)

Framework

Modules that provide a framework or other tools.

- @angular/material
- angular-actions
- angular-rules-engine

Library CONSUMERS

WHO WILL CONSUME YOUR MODULE?

FORMATS for different CONSUMERS

ESM2015

- Webpack (Optimized)
- Google Closure Compiler

ESM5 *

- CLI
- Rollup
- Webpack

UMD

- <script>
- Plunker
- Fiddle
- Node.js

* Angular Applications

Angular Package Format

HOW DO WE BUILD MODULES TARGETING DIFFERENT FORMATS?

Angular Package Format :: v6 (new)

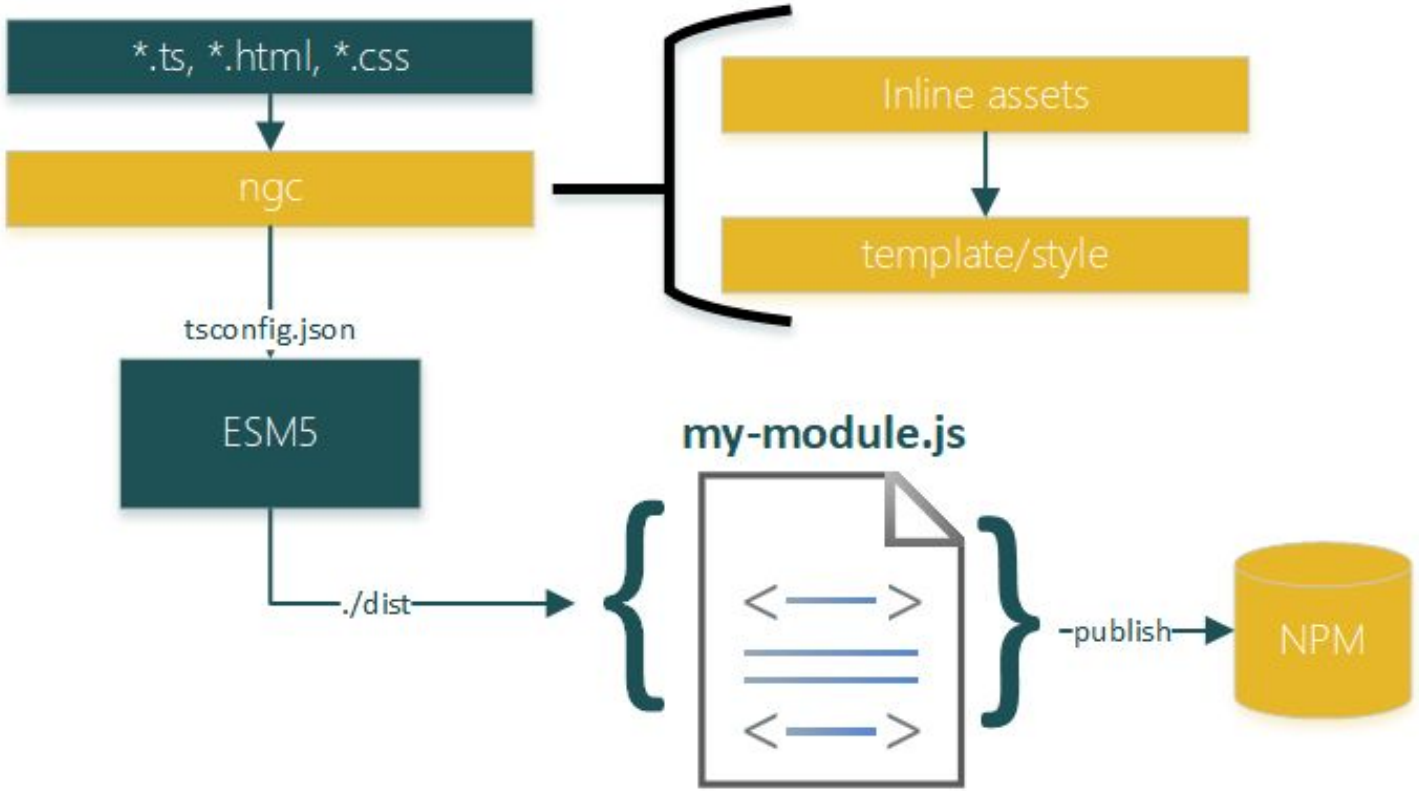
APF

This format applies to packages distributing Angular components (like Angular Material) as well as the core framework packages published under the `@angular` namespace, such as `@angular/core` and `@angular/forms`.

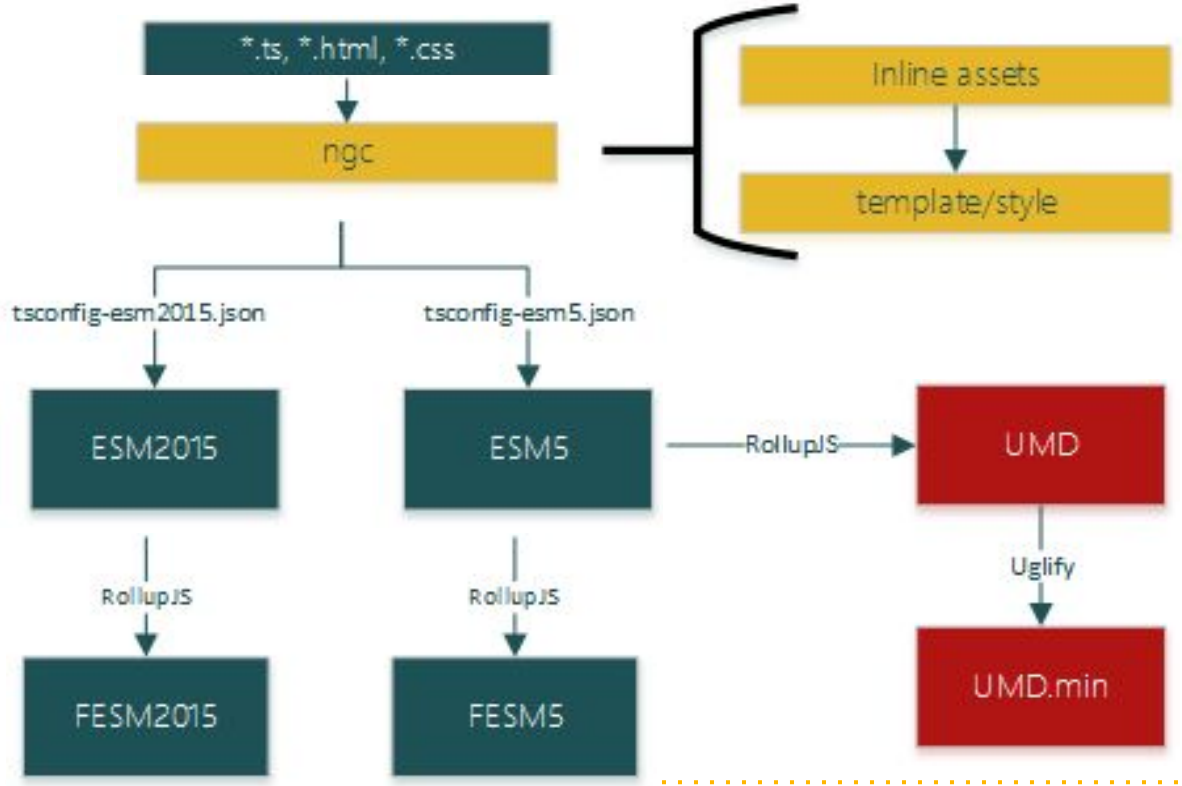
The format described here uses a distinct file layout and metadata configuration that enables a package to work seamlessly under most common scenarios where Angular is used, and makes it compatible with the tooling offered by the Angular team and the community itself.

USAGE Scenarios and ENVIRONMENTS

Build / Bundler / Consumer	Module Format	Primary Entry Point resolves to
WebPack (optimized) / Closure Compiler	ESM+ ES2015	`esm2015/index.js`
CLI / Rollup / WebPack	ESM+ES5	`esm5/index.js`
Plunker / Fiddle / ES5 / script tag	UMD	`Requires manual resolution by the developer to:` `bundles/core.umd.js and bundles/core.umd.min.js`
Node.js	UMD	`bundles/core.umd.js`


*.ts, *.html, *.css
ngc
Inline assets
template/style
tsconfig.json
ESM5
my-module.js
./dist
-publish
NPM

Build for CONSUMERS.



Use the FORCE Dude!

```
1 .\node_modules\.bin\ngc -p .\tsconfig.json
```

package.json

Property Name	Purpose
module	tools consuming ESM+ES5 (CLI, WebPack, Rollup)
es2015	property is used by tools consuming ESM+ES2015 (Closure, custom Webpack build)
main	Node.js
typings	typescript compiler (tsc)

{ package.json }

```
{
  "name": "@buildmotion/security",
  "version": "1.0.0",
  "description": "Add description here.",
  "main": "./bundles/buildmotion-security.umd.js",
  "module": "./esm5/buildmotion-security.js",
  "es2015": "./esm2015/buildmotion-security.js",
  "typings": "./buildmotion-security.d.ts"
}
```

DEMO Time

Let's build a library that can be published to npm...

ng-packagr

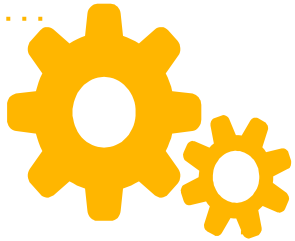
"Pat, I'd like to buy a vowel."

ng-packagr

Builds a library from Typescript sources and creates a distribution-ready npm package.

- Targets consumers using different module formats (see Angular Package Format)
- Publish package to npm

How to get STARTED?



- Install the package from npm.
- Add lib-specific package.json
  - Configure library
- Build library to destination
- Publish to npm

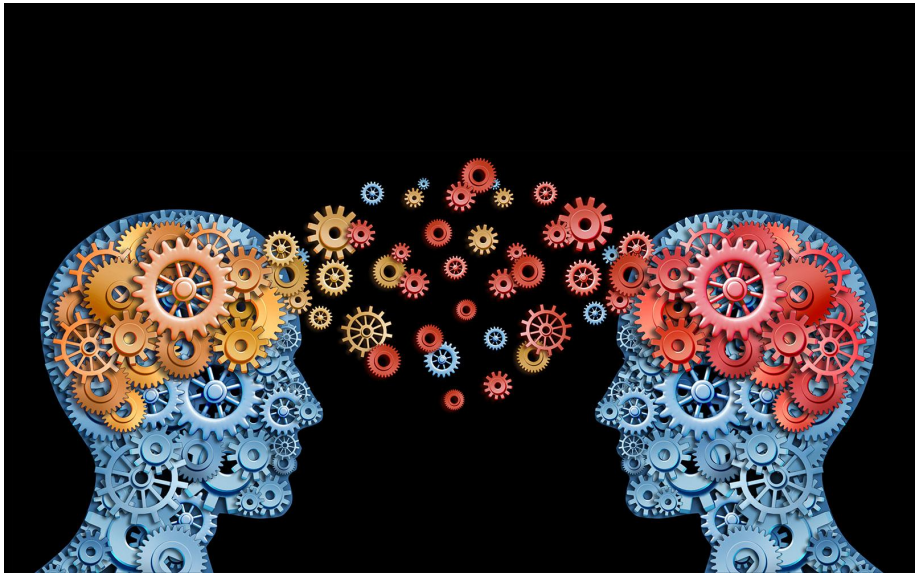
```
npm install --save-dev ng-packagr
```

package.json

```
"$schema": "./node_modules/ng-packagr/package.schema.json",
 "name": "@buildmotion/security",
 "version": "1.6.3",
 "description": "An Angular module to support OAuth authentication, login, subscriber, confirmations and resets.",
 "ngPackage": {
   "lib": {
     "entryFile": "./src/app/index.ts",
     "umdModuleIds": {
       "angular-rules-engine": "angularRulesEngine",
       "@buildmotion/logging": "buildmotionLogging",
       "@buildmotion/foundation": "buildmotionFoundation",
       "@buildmotion/core": "buildmotionCore"
     }
   },
   "dest": "./dist/@buildmotion/security"
 },
```

Why should I use this?

- Share your libraries with the community.
- Uses semantic versioning.
- Publish to public or private package repositories.
- Distribute within an organization.
- Become rich and famous and build rocket ships to Mars.



DEMO Time

Let's build a module that can be published to npm with support for different module formats.

Final THOUGHTS

The future is so bright...

REMEMBER WHY.

- Don't Repeat Yourself (DRY) Principle
- Separation of Concerns
- Single Responsibility
- Encapsulation
- Interface-Driven Design
- Versioning
- ...

BE SO GOOD
THEY CANNOT
IGNORE
YOU.

Presentation RESOURCES

- Angularlicious Guide: Custom Angular Modules
- Angularlicious Podcast
- Module Counts
- Angular Package Format v6
- Angular Core/Shared Modules
- Juri Strumphoner :: Github.com
- Ng-packagr
- ECMA Script 6 Modules
- Angular 6 - How to create libs/apps

Thanks!

Any questions?

You can find me at @angularlicious & matt@angularlicio.us