



Custom **LIBS** for Angular Apps

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



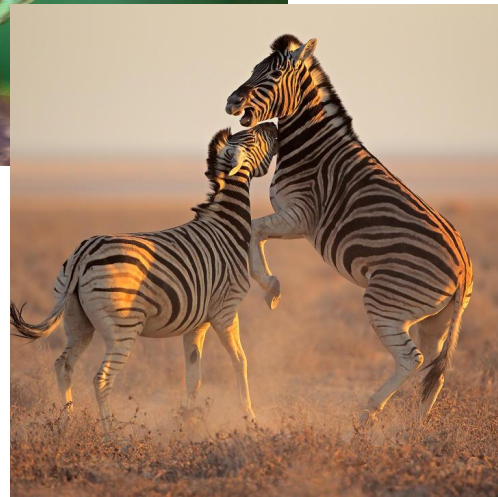
**In a galaxy far far
away...**

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?

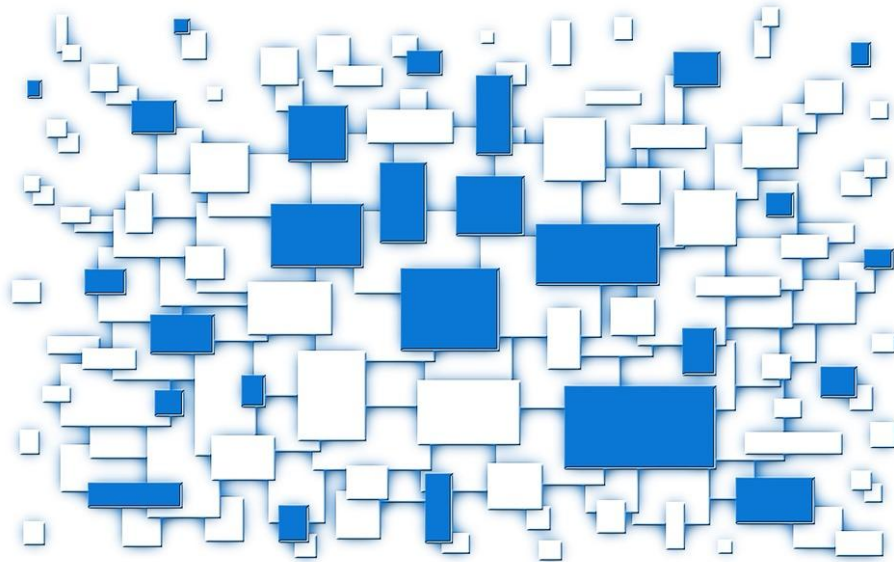
WHAT is A MODULE?



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?



How **MANY** MODULES?

598,876

Module Packages in NPM as of 3/14/2018

558 Average Growth

Per day

107,313/96

Compared to nuget (.NET)

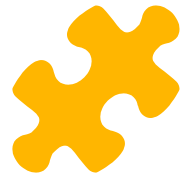


790+ Worldwide Meetups

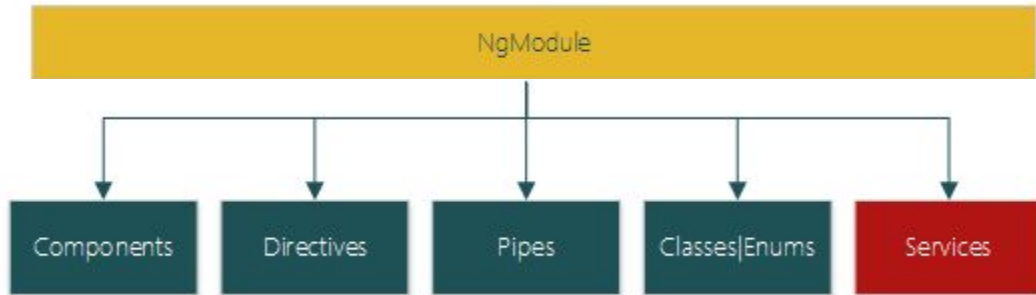


Module TYPES

Not all modules are the same...hmmm?



@NgModule Members



Different TYPES of MODULES



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



Module 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



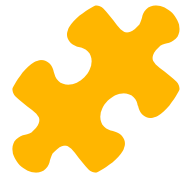
HOW?

HOW DO WE BUILD MODULES TARGETING DIFFERENT FORMATS?



Angular Package Format

GUIDELINES, BEST PRACTICES, AND FORMATS



Angular Package Format :: v6 (new)

APF

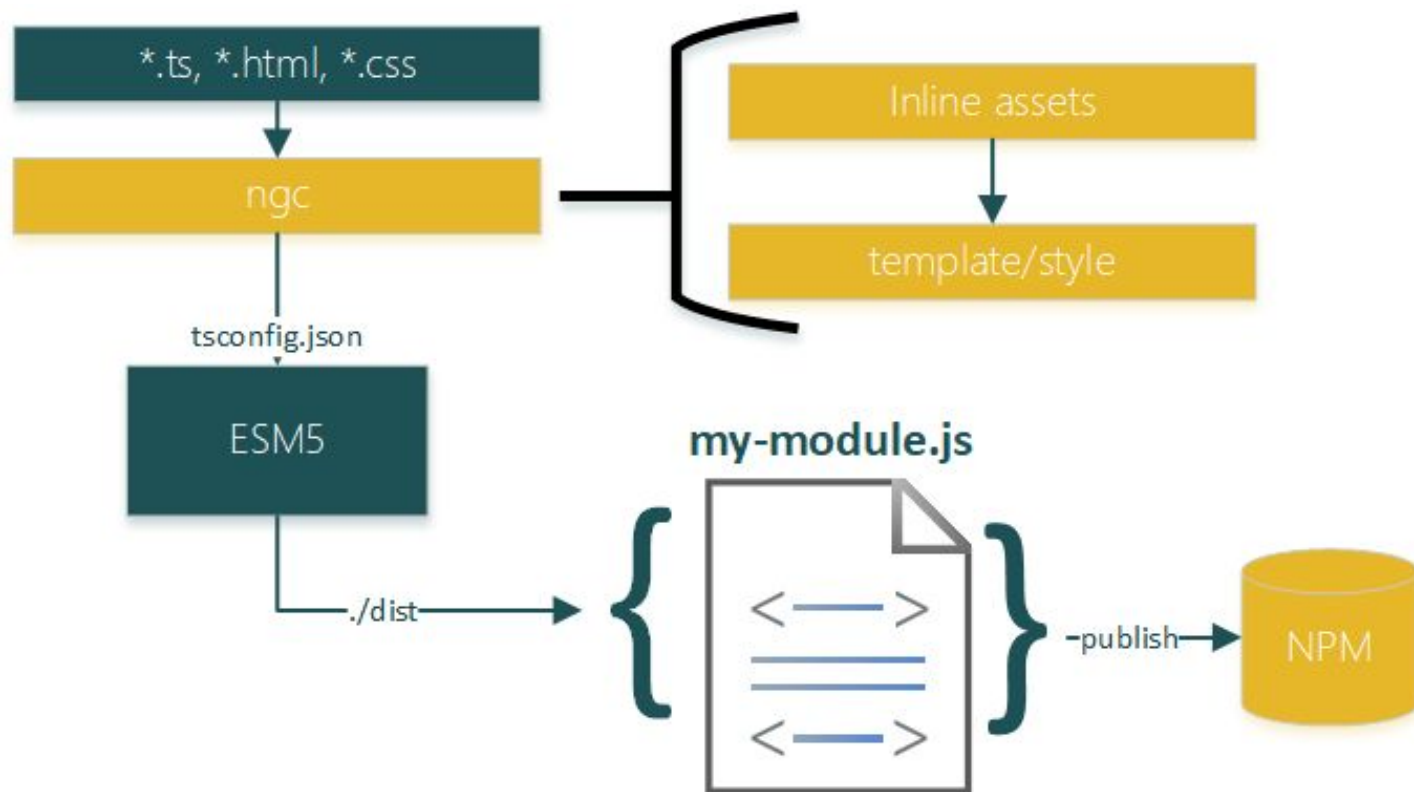
This format applies to packages distributing Angular components (like Angular Material) as well as the core framework packages published under the `@angular` namespace, such as `@angular/core` and `@angular/forms`.

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.

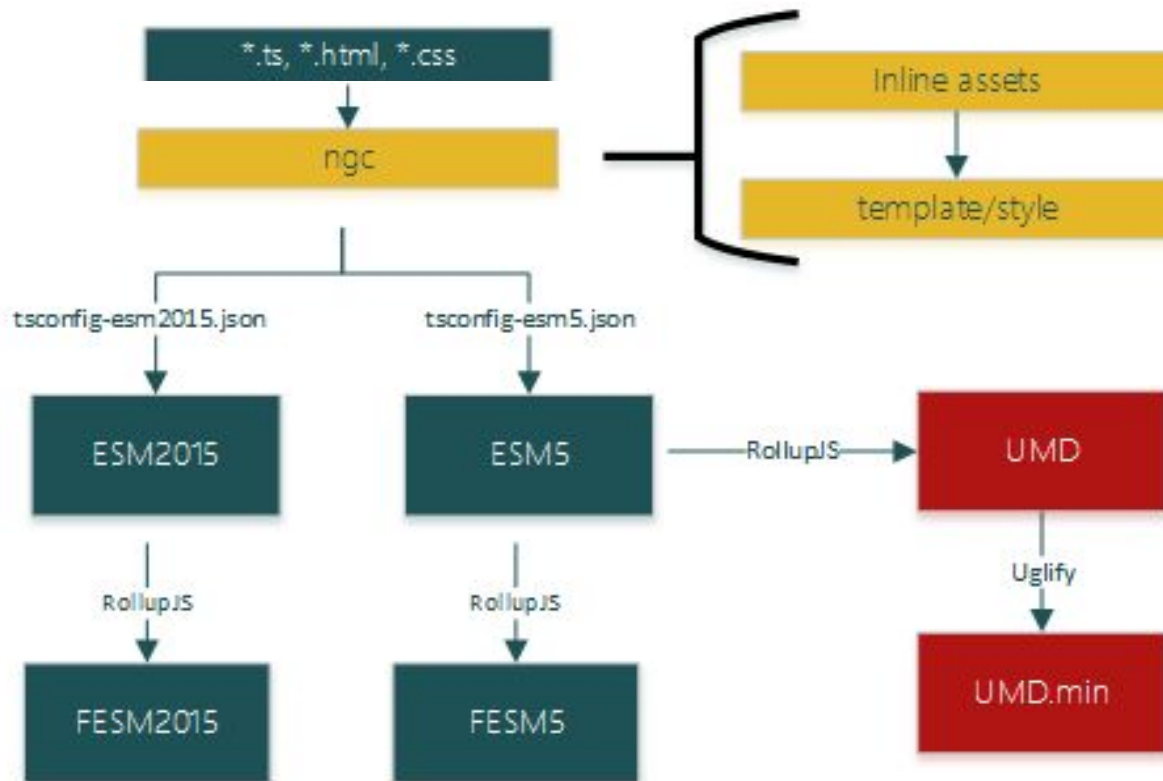


USAGE Scenarios and ENVIRONMENTS

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



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 }

```
1 {  
2   "name": "@buildmotion/security",  
3   "version": "1.0.0",  
4   "description": "Add description here.",  
5   "main": "./bundles/buildmotion-security.umd.js",  
6   "module": "./esm5/buildmotion-security.js",  
7   "es2015": "./esm2015/buildmotion-security.js",  
8   "typings": "./buildmotion-security.d.ts"  
9 }
```

DEMO Time

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



The future of **MODULES** and **TOOLS**.



Angular 6

There might be support from the Angular CLI to create modules using the APF.

Bazel

The future build tool for the modules might be Bazel - Google would like to support the same tool externally that they do internally.



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 Strumponer :: Github.com](#)
- [Ng-packagr](#)
- [ECMA Script 6 Modules](#)



Thanks!

Any questions?

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