



Atos

Global Knowledge.®

# Angular Module – Advanced Introduction

Peter Kassenaar –  
[info@kassenaar.com](mailto:info@kassenaar.com)

WORLDWIDE LOCATIONS

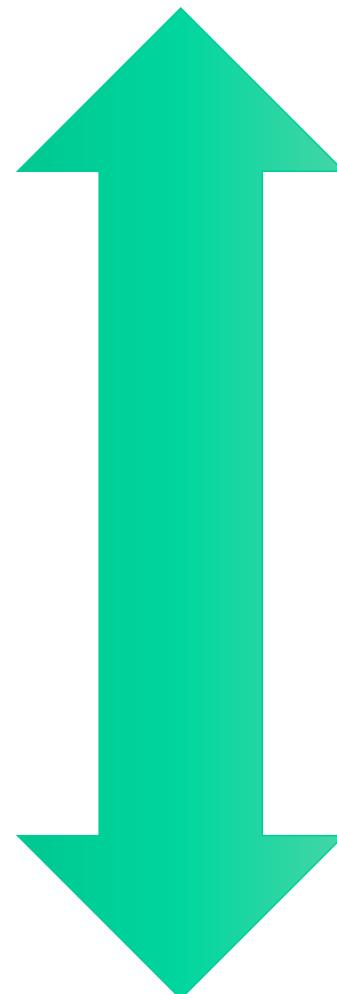
BELGIUM CANADA COLOMBIA DENMARK EGYPT FRANCE IRELAND JAPAN KOREA MALAYSIA MEXICO NETHERLANDS NORWAY QATAR  
SAUDI ARABIA SINGAPORE SPAIN SWEDEN UNITED ARAB EMIRATES UNITED KINGDOM UNITED STATES OF AMERICA

# Broadening?



or...

# deepening?



# Agenda - 2 days

- Various
  - Angular CLI tips & tricks
  - Angular Console
- Architecture: NG applications with multiple modules
  - Routing and Lazy loading modules
  - Loading strategies
  - Monorepo, micro-apps or not?
- Patterns:
  - Smart components/View components –
  - Content Projection
- Managing state with @ngrx/store
- Angular in the Enterprise, monorepo's, ...
- Angular and realtime databases/Firebase

# Labs and example code

## 1. Labs/Exercises

- In the PDF's in the Github-repo. But: feel free to deviate. Adapt to suit your own needs! (hobby, work, current projects)

## 2. Example code

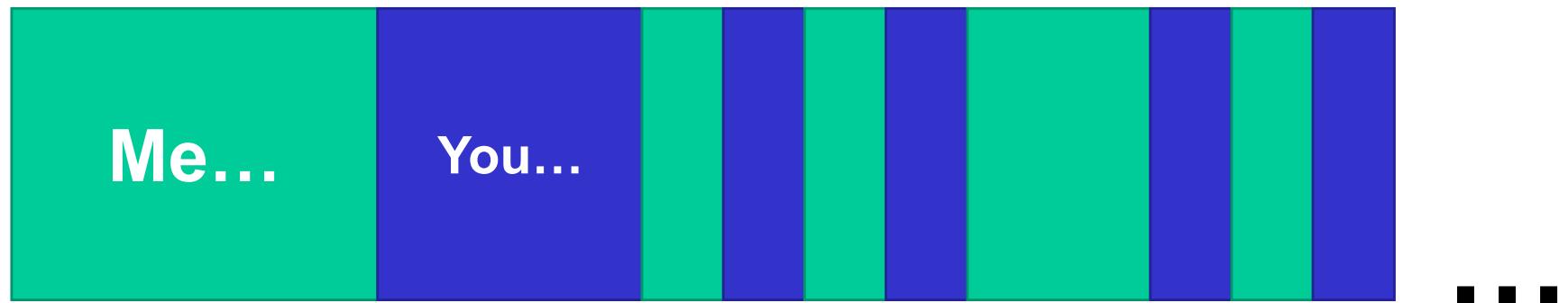
- Executions of the exercises, small projects (`npm install`, `npm start`)
- Work in progress – let me know of additions/errors!
- [github.com/PeterKassenaar/AngularAdvanced](https://github.com/PeterKassenaar/AngularAdvanced)

# Generic Github repo

The screenshot shows a GitHub repository page for [PeterKassenaar / AngularAdvanced](#). The page includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and a header with Unwatch (1), Star (3), Fork (8) buttons. Below the header are tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Security, Insights, and Settings. A main description states: "Labs, exercises and example code on the training Angular Advanced by Peter Kassenaar, info@kassenaar.com" with a link to <https://www.angulartraining.nl/>. There are buttons for Edit, angular, training, and Manage topics. Key statistics shown are 135 commits, 1 branch, 0 releases, and 2 contributors. A pull request button and file management buttons (Create new file, Upload files, Find file, Clone or download) are also present. The commit history lists the following changes:

File	Description	Time
<a href="#">examples</a>	Updated examples to Angular V8	2 days ago
<a href="#">.angulardoc.json</a>	Updated angular.json	last year
<a href="#">.gitignore</a>	Update .gitignore	2 years ago
<a href="#">README.md</a>	Update README.md	27 days ago

# Overall process

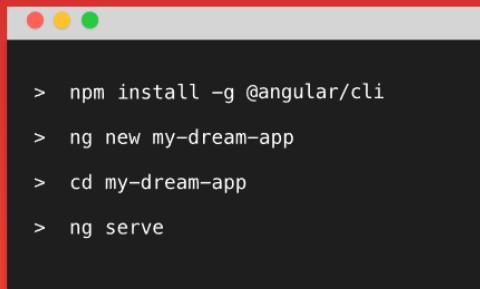


# Questions?



# Angular CLI

Scaffold new projects, modules, components via command line...



```
> npm install -g @angular/cli
> ng new my-dream-app
> cd my-dream-app
> ng serve
```

## Angular CLI

A command line interface for Angular

[GET STARTED](#)

## ng new

The Angular CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

## ng generate

Generate components, routes, services and pipes with a simple command. The CLI will also create

# We'll be using Angular-CLI this course

- It *is* possible to configure your Angular app by hand
- Using the CLI it's much simpler.
- CLI-options:
  - Scaffolding
  - Generating
  - Testing
  - Building
  - AOT-Compiling
  - ...

<https://cli.angular.io>

➤ README.md

# Angular CLI

[chat on gitter](#)

[build passing](#) [dependencies up to date](#) [devDependencies up to date](#) [npm v1.2.0](#)

CLI for Angular applications based on the [ember-cli](#) project.

## Note

The CLI is now in 1.0. If you are updating from a beta or RC version, check out our [1.0 Update Guide](#).

If you wish to collaborate, check out [our issue list](#).

Before submitting new issues, have a look at [issues marked with the type: faq label](#).

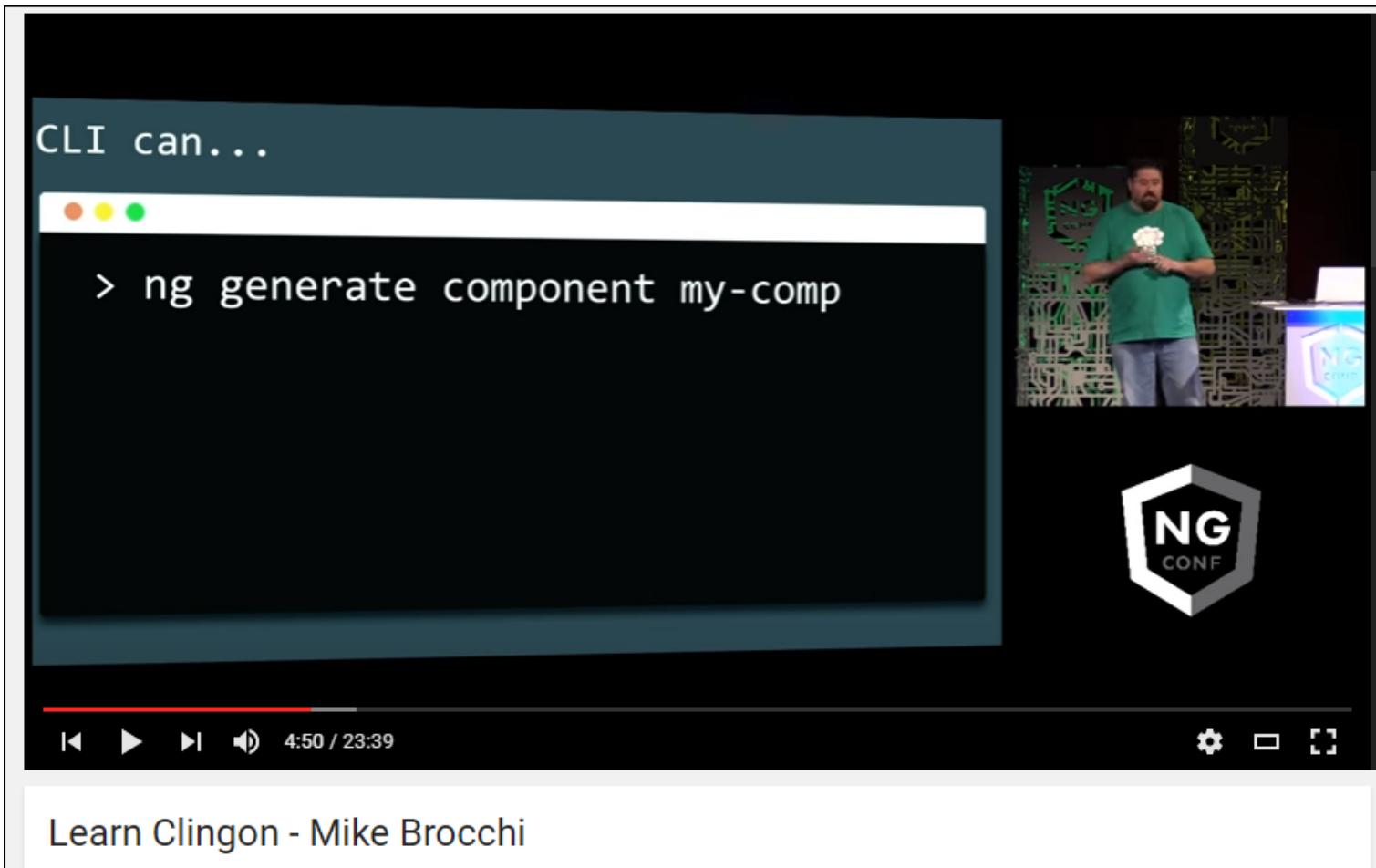
## Prerequisites

Both the CLI and generated project have dependencies that require Node 6.9.0 or higher, together with NPM 3 or higher.

## Table of Contents

- [Installation](#)
- [Usage](#)
- [Generating a New Project](#)
- [Generating Components, Directives, Pipes and Services](#)
- [Updating Angular CLI](#)

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



<https://www.youtube.com/watch?v=wHZe6gGI5RY>

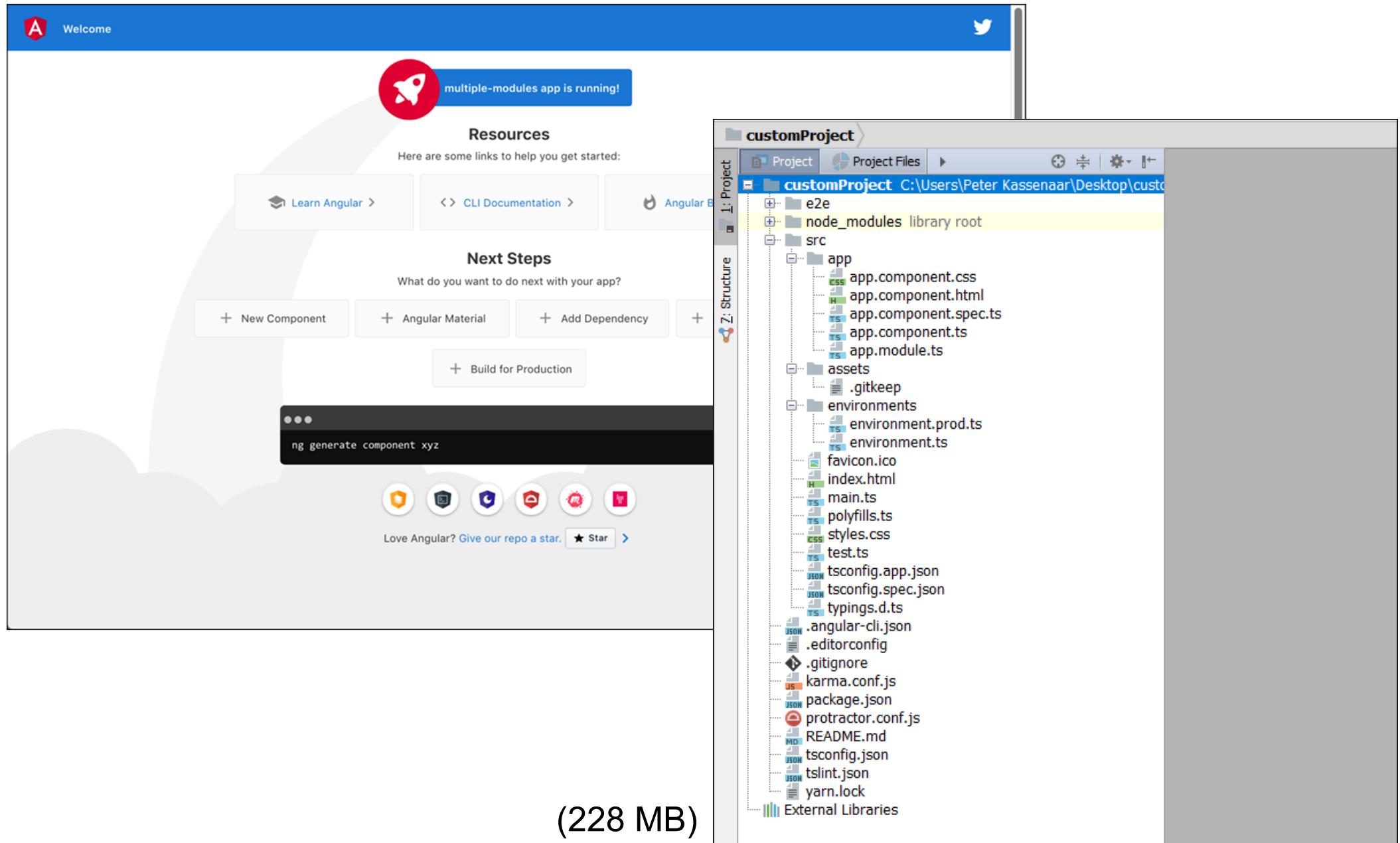
# Main commands

## **ng new – create basic app**

```
ng new PROJECT_NAME  
cd PROJECT_NAME  
ng serve
```

Project is served on `http://localhost:4200`

# Default application



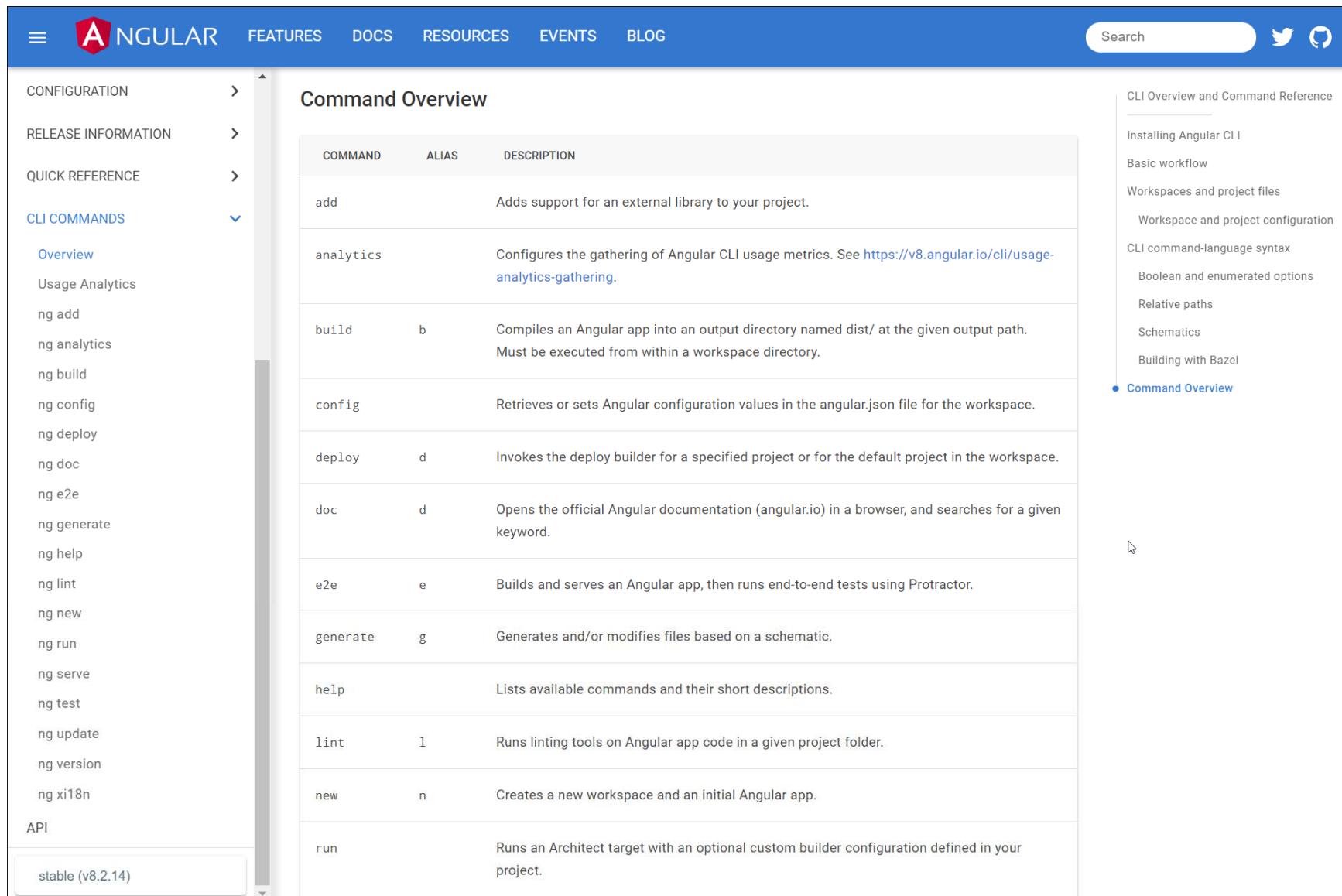
# Some CLI tips & tricks

- `ng serve --open` Directly open the compiled project in the browser
- `ng serve --port 4300` Serve project on different port
- `ng serve --ssl` Serve using `https://`
- `ng serve --live-reload false` Do not use live reload
- `ng serve --help` Overview of all other options

## More `ng` tooling

- `ng generate <blueprint> --dry-run` Do not write output files
- `ng generate <blueprint> --spec false` Do not write spec file
- `ng generate module <name> --routing` add routing to new module

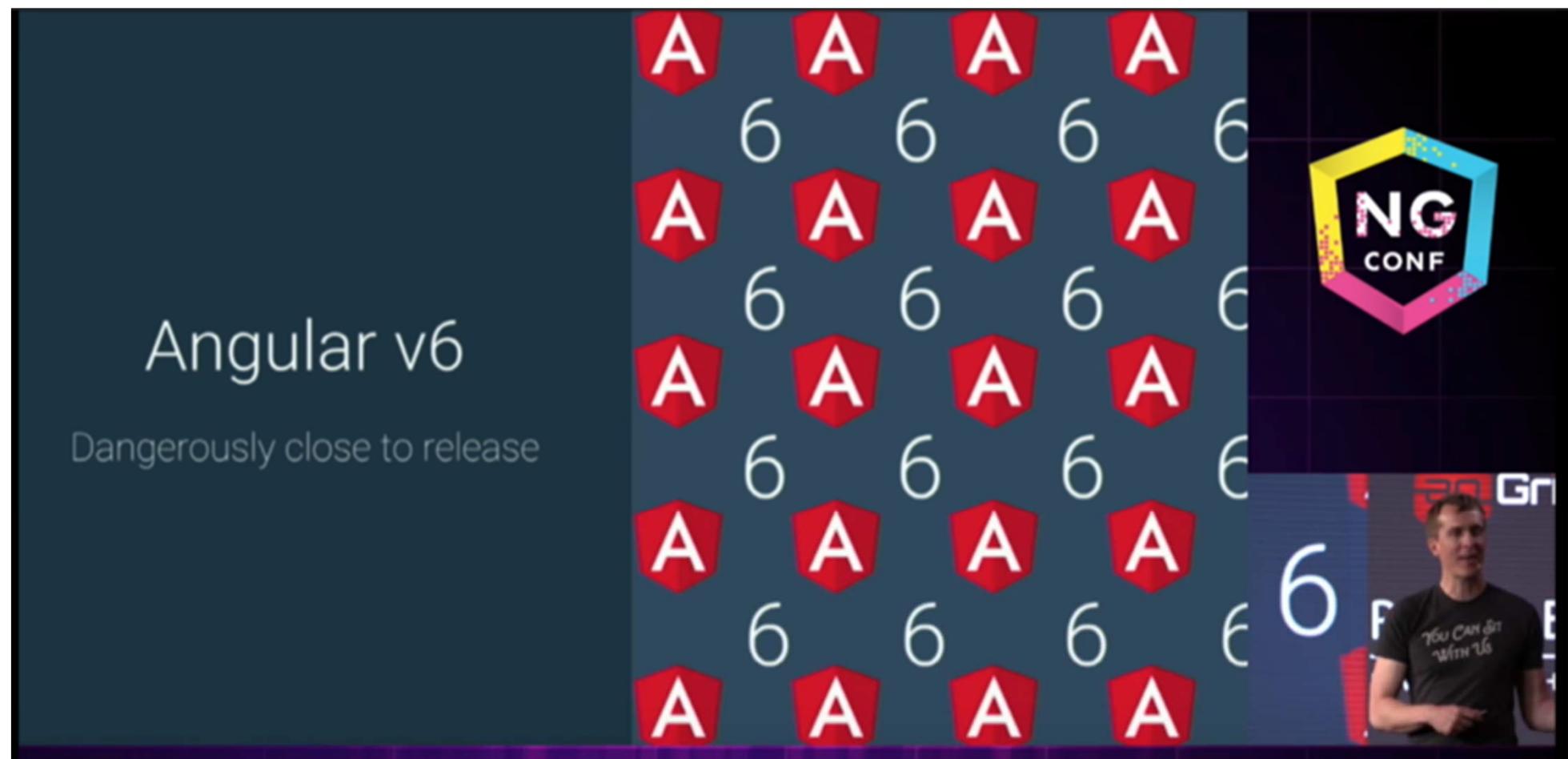
# Lots (!) of options



The screenshot shows the Angular CLI documentation page for 'Command Overview'. The left sidebar has sections for Configuration, Release Information, Quick Reference, and a expanded 'CLI COMMANDS' section containing links for Overview, Usage Analytics, and various ng commands like ng add, ng build, etc. A 'stable (v8.2.14)' note is at the bottom. The main content area has a table with columns for COMMAND, ALIAS, and DESCRIPTION. The table lists 14 commands: add, analytics, build, config, deploy, doc, e2e, generate, help, lint, new, serve, test, update, version, and xi18n. The 'Command Overview' link in the sidebar is highlighted with a blue dot.

COMMAND	ALIAS	DESCRIPTION
add		Adds support for an external library to your project.
analytics		Configures the gathering of Angular CLI usage metrics. See <a href="https://v8.angular.io/cli/usage-analytics-gathering">https://v8.angular.io/cli/usage-analytics-gathering</a> .
build	b	Compiles an Angular app into an output directory named dist/ at the given output path. Must be executed from within a workspace directory.
config		Retrieves or sets Angular configuration values in the angular.json file for the workspace.
deploy	d	Invokes the deploy builder for a specified project or for the default project in the workspace.
doc	d	Opens the official Angular documentation (angular.io) in a browser, and searches for a given keyword.
e2e	e	Builds and serves an Angular app, then runs end-to-end tests using Protractor.
generate	g	Generates and/or modifies files based on a schematic.
help		Lists available commands and their short descriptions.
lint	l	Runs linting tools on Angular app code in a given project folder.
new	n	Creates a new workspace and an initial Angular app.
run		Runs an Architect target with an optional custom builder configuration defined in your project.

# Angular CLI 6.0 - As of May 2018



<https://www.youtube.com/watch?v=dIxkngPOWms>

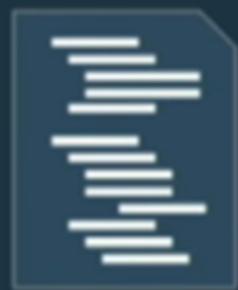
# About version numbering

## Aligning Library Releases

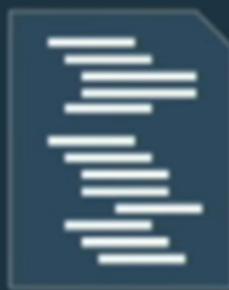
	Today	With v6
Angular	5.2.10	6.0
Material	5.2.4	6.0
CLI	1.7	6.0

# New CLI Options

Extending the CLI with Schematics



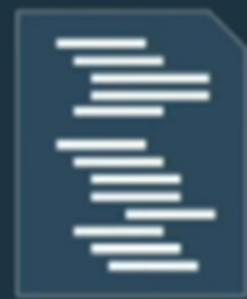
new



generate

component  
directive  
pipe  
service  
...

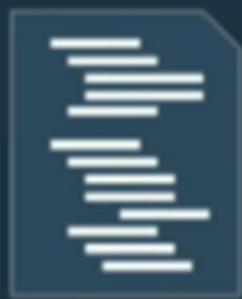
# Extending the CLI with Schematics



new



generate



update

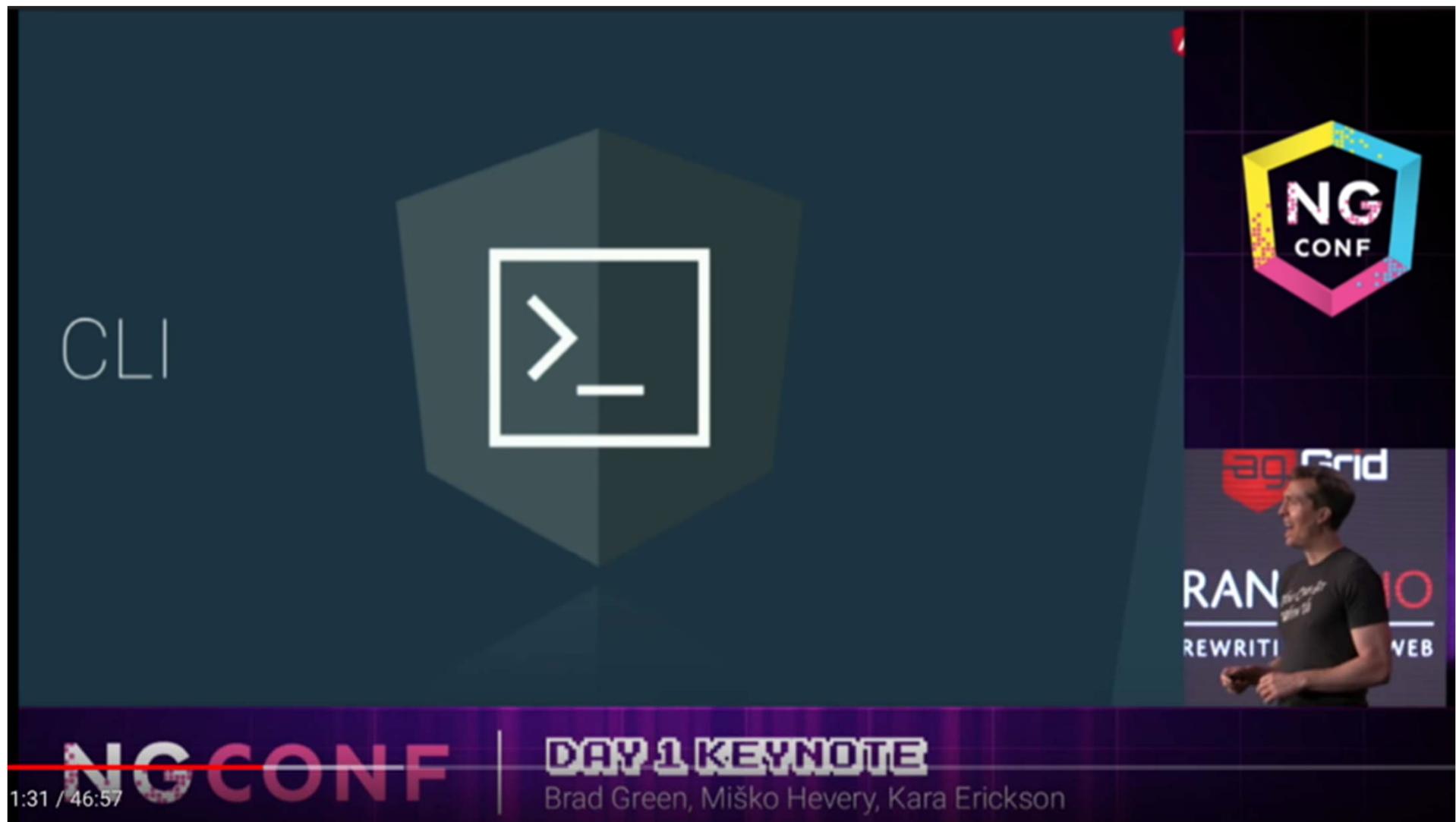


add

component  
directive  
pipe  
service

...

# Info on the Angular 6.x keynote



<https://www.youtube.com/watch?v=dIxknqPOWms>

# Ng add overview

The screenshot shows the Angular CLI's `ng add` feature page. At the top, there's a terminal window showing the command sequence: `> npm install -g @angular/cli`, `> ng new material-app`, `> cd material-app`, and `> ng add @angular/material`. To the right, the main heading is **NG ADD** with the subtitle "Supercharge Your Angular Project". A "Submit Request!" button is visible. Below this, four cards are displayed:

- Progressive Web Apps** (@Angular) - Adds PWA features to new or existing Angular applications. Command: `ng add @angular/pwa`.
- Web Components** (@Angular) - Creates native web components using Angular. Command: `ng add @angular/elements`.
- Material Design** (@Angular) - Adds Angular Material to your App and enjoys a growing list of beautiful components.
- Bootstrap Framework** (@NG-Bootstrap) - Angular components collection built from the ground up using Bootstrap 4.

<https://ngadd.com/>

# Angular Console – As of August 2018

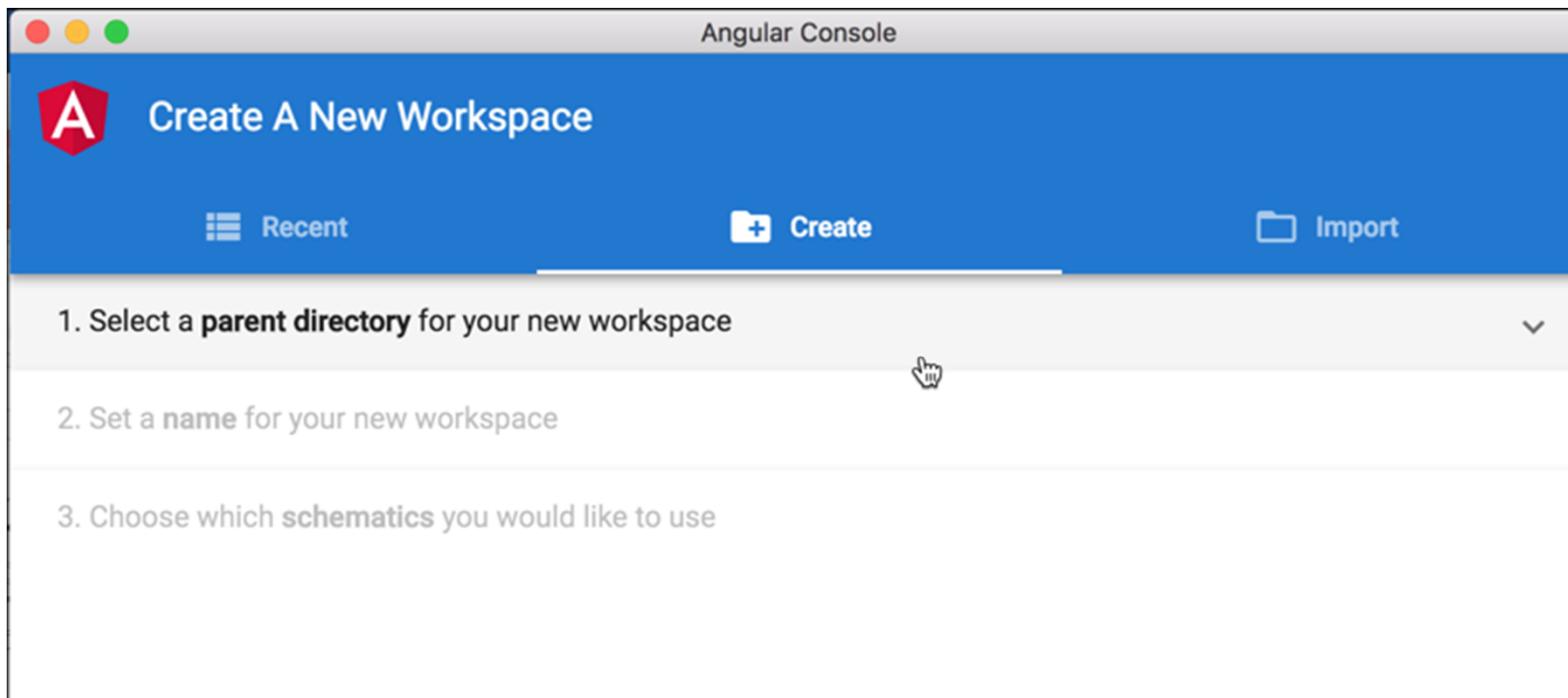
The screenshot shows the Angular Console website with a blue header bar. On the left is a logo icon of a computer monitor with a red 'A' on it, followed by the text 'Angular Console'. On the right are three navigation links: 'Download', 'About', and 'Features'. Below the header, the main title 'The Power of the Angular CLI.  
The Convenience of an App.' is displayed in large white font. A subtitle below it reads 'Spend less time looking up command line arguments, and more time shipping incredible products.' A prominent red button with the text 'Download Angular Console' is centered. At the bottom, there's a white sidebar featuring a browser icon and the text 'Angular Console'.

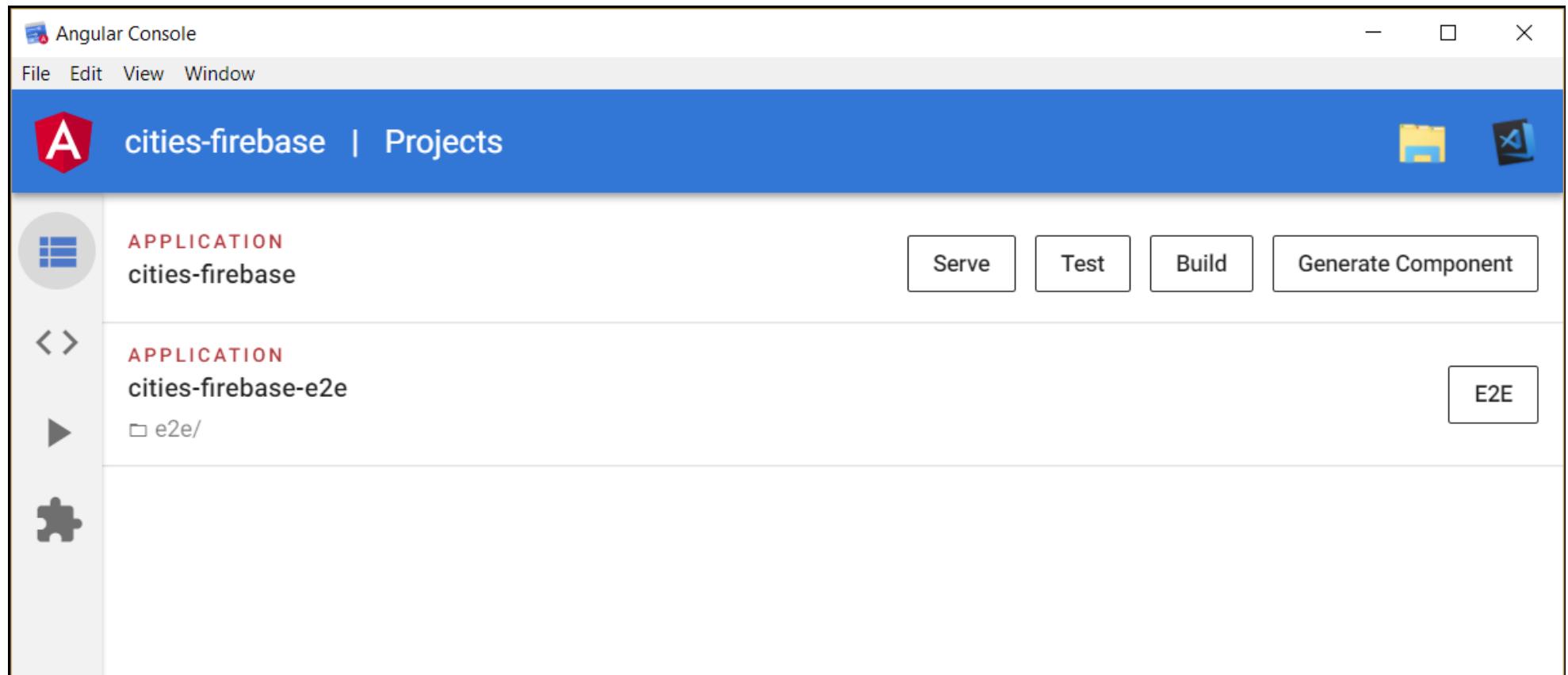
<https://angularconsole.com/>

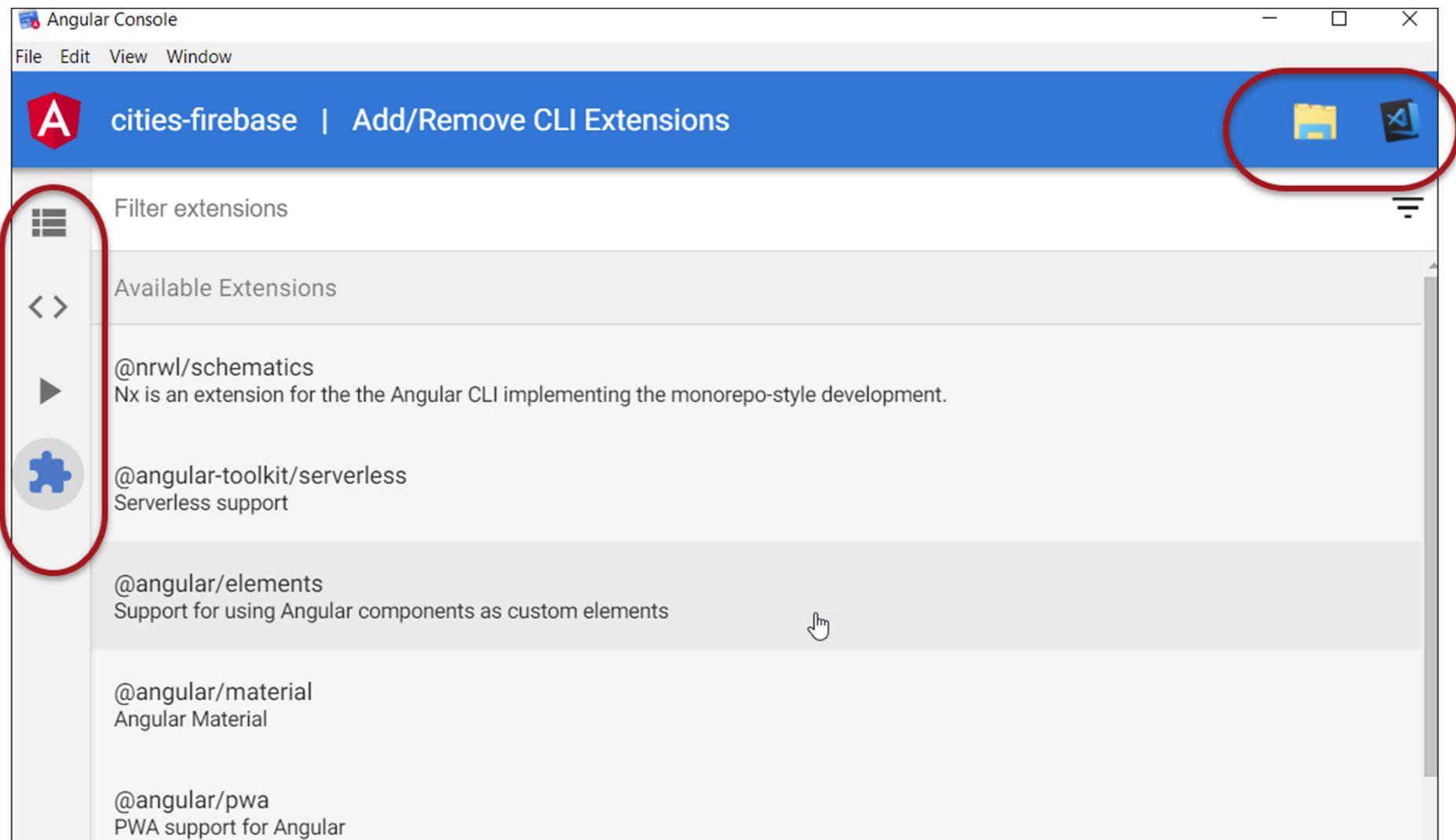
# Angular Console

- By nrwl.io, team of Victor Savkin and Jeff Cross
- Visual interface to the CLI
  - Code generation
  - Run Custom NPM scripts
  - Discover and install extensions
  - Build CLI commands visually – no need to remember all shortcuts
  - Integrated terminal output
  - Create new projects, import existing projects

*“A Gateway for  
Beginners. A Powertool  
for Experts.”*







# More background information

≡ ↻ sitepoint

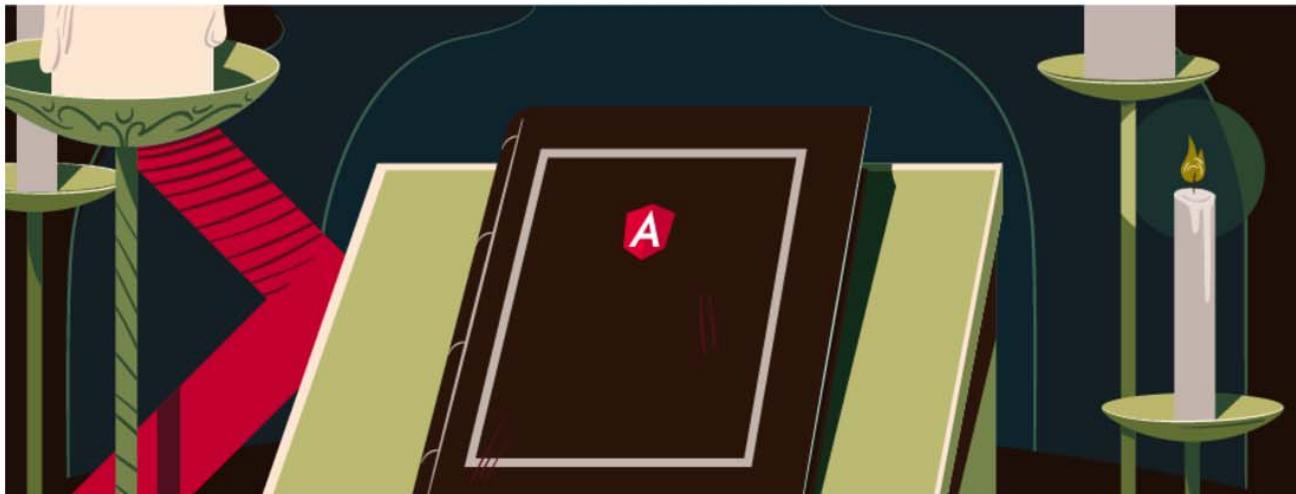
Courses Books Community Login Create Free Account

HTML & CSS JAVASCRIPT PHP DESIGN & UX WEB WORDPRESS SEO BY WOORANK MICROSOFT TECH

JavaScript Article By Jurgen Van de Moere April 25, 2017

## The Ultimate Angular CLI Reference Guide

Microsoft - helping keep SitePoint free!



RECOMMENDED

- ① Managing State in Aurelia: How to Use Aurelia with Redux
- ② Create Your Own Yeoman-Style Scaffolding Tool with Caporal.js
- ③ 5 Time-Saving Uses for WP-CLI Automation
- ④ Easy AngularJS Authentication with Auth0

<https://www.sitepoint.com/ultimate-angular-cli-reference/>

# Custom Angular Schematics

The image shows a screenshot of a Medium article page. At the top left is a black 'M' logo. To the right are a search icon, a green circular notification badge with the number '6', an 'Upgrade' button, and a user profile picture. The main title 'Total Guide To Custom Angular Schematics' is centered above a bio section. The bio includes a small circular profile picture of a man, the name 'Tomas Trajan', a 'Follow' button, and the date 'Mar 26 · 20 min read'. Below the bio is a large, high-angle aerial photograph of a road through agricultural fields. Overlaid on the bottom left of the photo is a large red hexagon containing a white capital letter 'A'. At the very bottom left, the words 'THE TOTAL GUIDE' are written in large, bold, white capital letters.

<https://medium.com/@tomastrajan/total-guide-to-custom-angular-schematics-5c50cf90cdb4>

# Angular CLI Builders

Add your own CLI features and augment existing ones:

<https://blog.angular.io/introducing-cli-builders-d012d4489f1b>



## Introducing CLI Builders



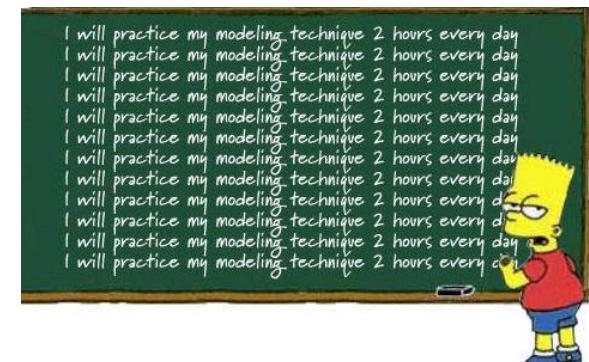
Hans [Follow](#)  
Apr 24 · 8 min read

In this blog post, we're going to look at a new API in Angular CLI, which allows you to add CLI features and augment existing ones. We'll discuss how to interact with this API and what are the extension points which allow you to add additional features to the CLI.

*You can find the code from the examples below in [this GitHub repository](#).*

# Workshop

- Download and install Angular Console
- Generate a new project with it
- *OR:*
- Import an existing project
  
- Generate a new component or a new service with it
- Run some scripts from within Angular Console (`start`, `build`, `serve`)
- Add some new CLI extensions, see **ngadd.com**; for instance
  - `@angular/material`
  - `@angular/elements`
  - See how/where they are installed



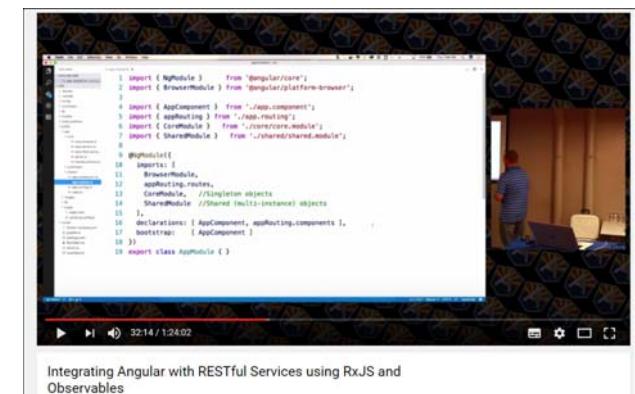


# Multiple modules

Splitting your application into separate, reusable modules

# Angular Modules

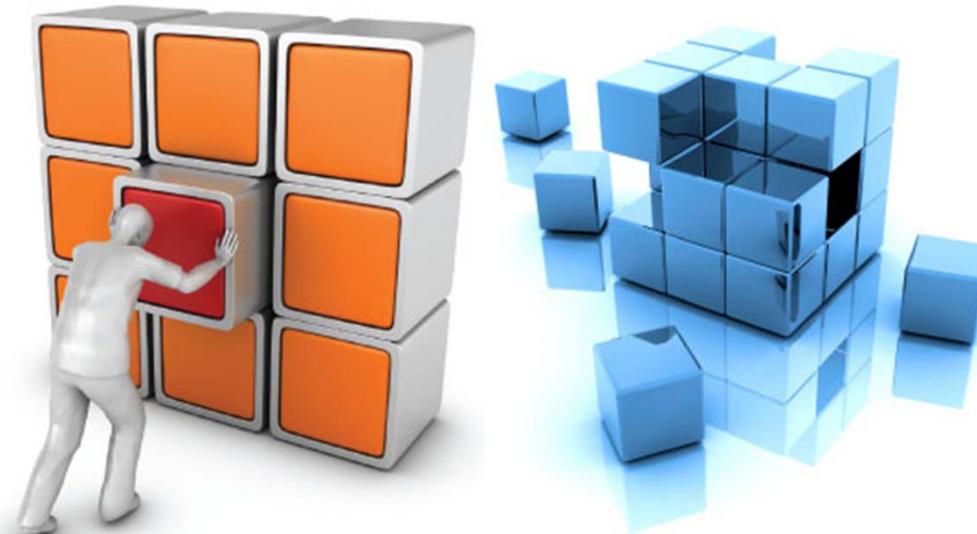
- Divide your app into *logical* and often *reusable* pieces of code
- Keyword : **code organization**
- Use one AppModule - the root of your app
- Use one CoreModule - containing all *singletons* in your app
- Use one SharedModule - containing all shared resources, possible multiple instances
- Use additional modules *per feature*
- <https://www.youtube.com/watch?v=YxK4UW4UfCk>

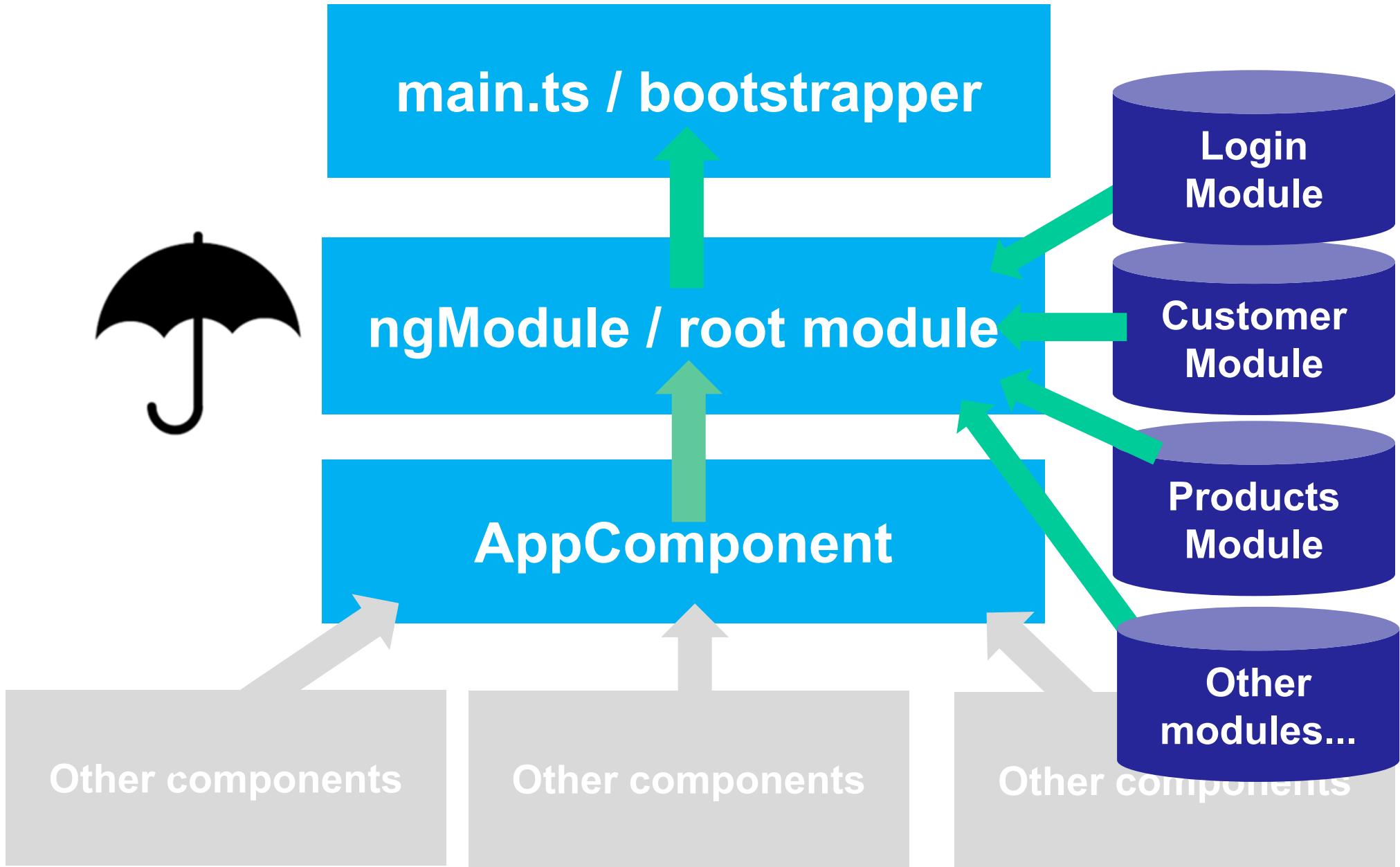


Integrating Angular with RESTful Services using RxJS and Observables

# Application – multiple Modules

- *Reuse* of Components, Pipes, Routes and Services etc. over different apps
- *Wrap* each set of logical related components, services, etc. in its own module.





# Steps

## 1. Create a new module

- Optional: test first with `--dry-run`
- `ng generate module customers --dry-run`

## 2. Create component(s) inside that module

- Again: test first with `--dry-run`
- `ng generate component customers --module customers --dry-run`

## 3. Apply UI, logic, etc. to your component

## 4. Export your component inside `customer.module.ts`

- `exports : [CustomerComponent],`
- Otherwise it can't be used in other components!

## 5. Provide new module to `app.module.ts`

- `imports: [CustomerModule]`

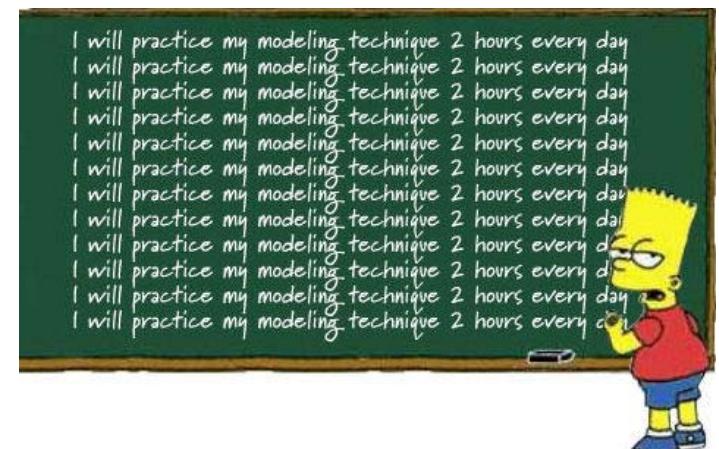
# Optional : SharedModule

- Reuse components in multiple modules? Use a SharedModule
  - `ng g m shared` – shorthand notation
- Create components inside SharedModule
- Import SharedModule in other modules
- It doesn't have to be in AppModule if you don't use it directly!
- It *does* not add size to module bundles



# Workshop

- Open ../100-multiple modules.
  - Create a new module
  - Create a new component inside this new module and give it some UI.
  - Include the module in the Main Module and show it besides other modules
  - Include the Search Component in your own module
- 
- *OR:*
  - Add Multiple Modules from scratch to your own application, using the steps described in this module.



# How to structure feature modules



242



## Why and how to structure Features in Modules in Angular

This might sound pretty basic, but I encounter these challenges over and over in customer projects and it's still an ongoing discussion internally.

A central project goal in a recent Angular project was to design features and UI components for reusability. To achieve this, we need to make sure our code is well isolated and has a simple and clear dependency model.

### Prologue: Feature vs. Technical Project Structure

When building small apps and looking at common code samples in the internet a lot of devs (including myself) tend to come up with a project structure like this:

```
MYAPP
  src
    app
      components
        home
          home.component.html
          home.component.ts
        user
          user.component.html
```

<https://medium.com/@philippbauknecht/why-and-how-to-structure-features-in-modules-in-angular-d5602c6436be>