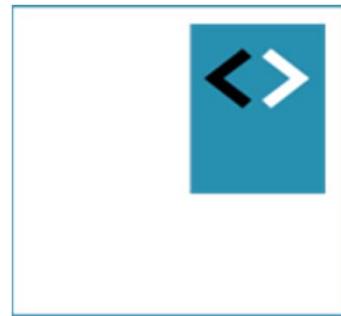




Angular Advanced

Using multiple Modules



Peter Kassenaar
info@kassenaar.com

Hoe ziet een opleiding er uit?



Symptomen? Blijf thuis!

Heb je symptomen die mogelijk duiden op CoViD-19, blijf zeker thuis. Indien je op de opleiding verschijnt met symptomen, word je naar huis gestuurd.



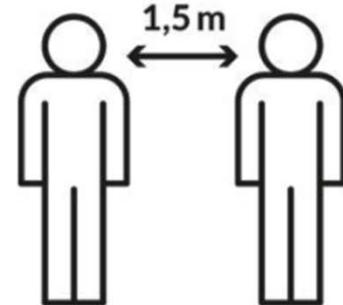
Mondmaskers verplicht

Het gebruik van mondmaskers is verplicht. Enkel waar voldoende afstand gegarandeerd kan worden, is het dragen van een mondmasker optioneel. Neem deze dus zeker mee naar de opleiding.



Handhygiëne

Voor de les begint en voor de lunchpauze ben je verplicht om je handen te wassen. Er is voldoende water en zeep en/of handgel beschikbaar.



Afstand

Elke opleiding vindt plaats in een locatie die ruim genoeg is om voldoende fysieke afstand te kunnen houden. Hou altijd minstens **1,5 meter afstand**.



Blijf van je gezicht

Raak je gezicht niet aan en vermijd elk fysiek contact.



Volg de instructies

De lesgever informeert je over de **te volgen richtlijnen**. Luister altijd naar zijn/haar instructies. Deze dienen om jouw veiligheid en die van de lesgever te garanderen.



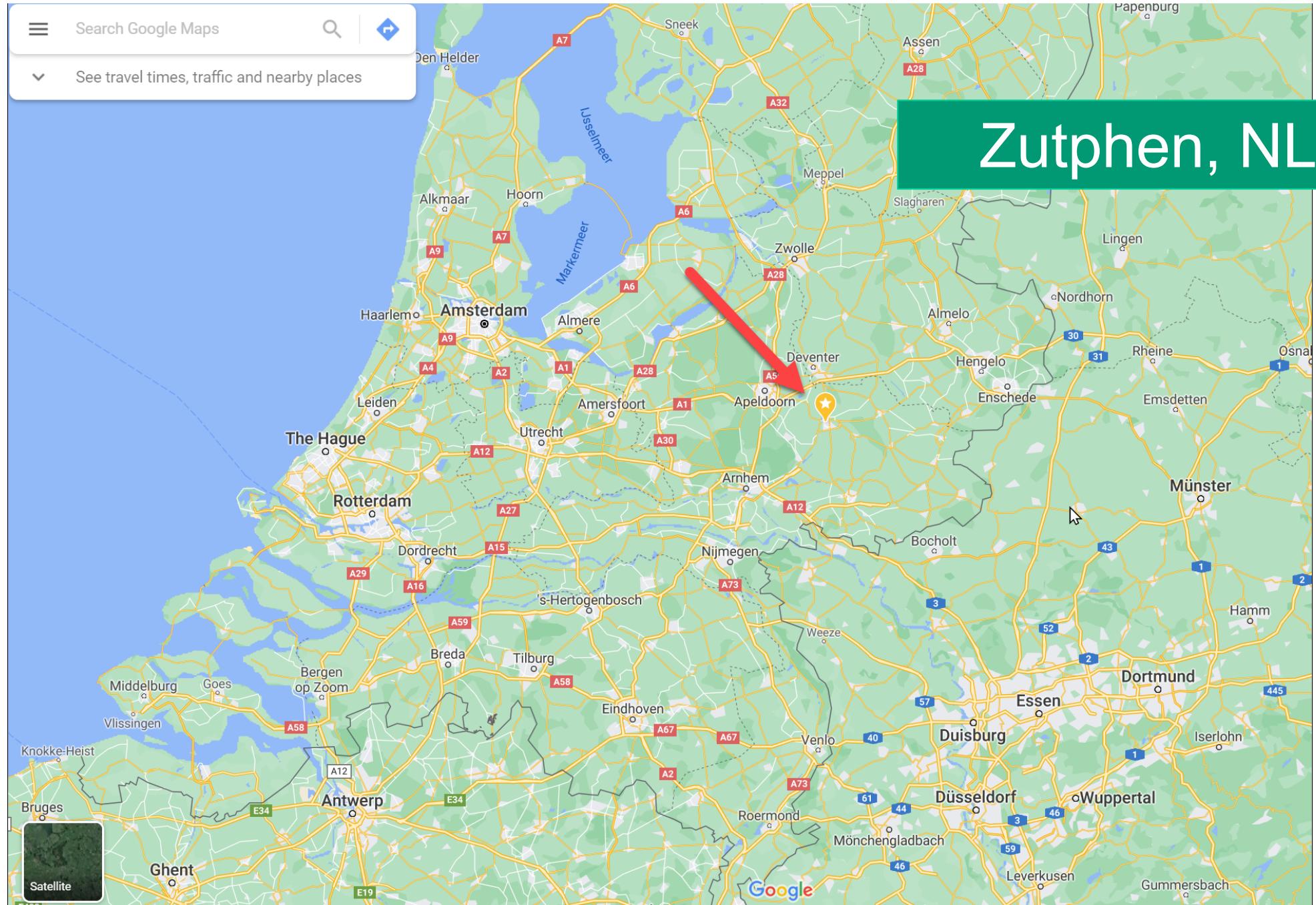
Materiaal

Al het materiaal dat je gebruikt is op voorhand **gereinigd** en wordt ook na elke les opnieuw gereinigd. Tijdens praktijklessen moet je steeds met (eigen) handschoenen werken.



Reiniging lokalen

Na elke les worden lessafels, stoelen, klinken en eventueel lesmateriaal opnieuw gereinigd.



Peter Kassenaar

- Trainer, author, developer – since 1996
- Specialty: “*Everything JavaScript*”
- JavaScript, ES6, Angular, NodeJS, TypeScript, jQuery, Vue.js, React

www.kassenaar.com

info@kassenaar.com



Twitter: [@PeterKassenaar](https://twitter.com/@PeterKassenaar)



Angulartraining.nl

Home Training Dates Information Contact

2018 dates now available!

```
const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', loadChildren: './home/home.module#HomeModule' },
  { path: 'training', loadChildren: './training/training.module#TrainingModule' },
];
const config: ExtraOptions = {
  enableTracing: false,
  preloadingStrategy: PreloadAllModules
};
@NgModule({
  imports: [RouterModule.forRoot(routes, config)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```



World-class Angular training in Dutch and English

Live classrooms - focused on today's developers

LEARN MORE SIGN UP!

www.angulartraining.nl

github.com/PeterKassenaar/aarixa

The screenshot shows a GitHub repository page for the user PeterKassenaar with the repository name aarixa. The page has a dark theme.

Header: Search or jump to... / Pull requests Issues Marketplace Explore

Repository Information: PeterKassenaar / aarixa Public Unwatch 1 Star 0 Fork 0

Navigation: Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Code View: main branch 1 branch 0 tags Go to file Add file Code

Commits:

File	Message	Commit Hash	Time	Commits
PeterKassenaar Update README.md		b2e6d5a	2 minutes ago	2 commits
.gitignore	Initial commit		3 minutes ago	
README.md	Update README.md		2 minutes ago	

About: Slides and example code on the training Angular - November 2021

Readme:

Releases: No releases published Create a new release

Packages: No packages published Publish your first package

Links:

- Generic repo with example code: <https://github.com/PeterKassenaar/voorbeeldenAngular2>
- ...

About you...



Introduce yourself shortly

Knowledge of Angular, (mobile/web-) apps?

How long have you worked with Angular yet?

Tell us a little bit about your projects.

What are your expectations of this course?

Agenda

17 - 18 Jan. 2022 – Mo.- Tue.

~9:00 start

~ 10:30 Break

~12:00 lunch

~ 14:30 Break

~16:00-16:15 closing

Material

- Software (Angular + Editor + Browser + libraries)
- Handouts (PDF, Github)
- Workshops (in the presentations)
- Websites (online)



angular.io/

Why devs and enterprises like Angular...



CLI



Router



HTTP



Forms



Animations



i18n



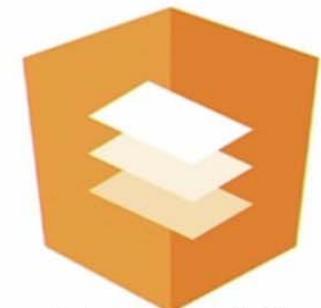
Testing



Language
Services



Universal



Material &
CDK

Short recap

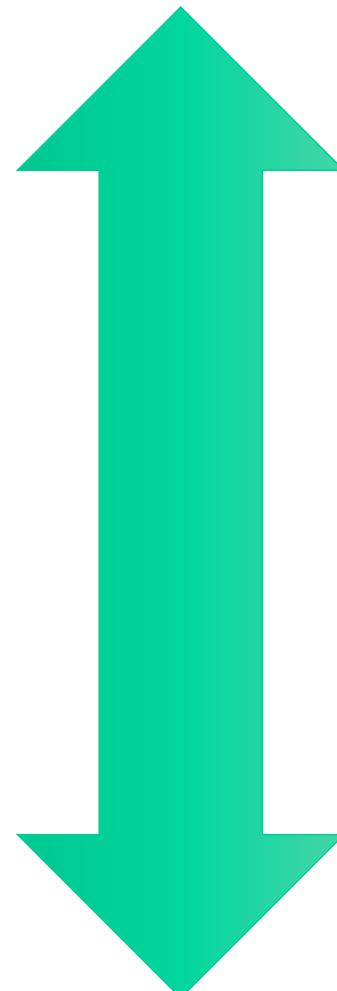
- November 2021: **Fundamentals**
 - Concepts, context & architecture
 - Angular CLI basics
 - Components, Data binding
 - Services
 - Live API's
 - Component communication / event buses
 - Routing
 - Basics
 - Routing Parameters

Broadening?



or...

deepening?



Agenda - 2 days - Thematic



- Day 1: Architecture + Adv. Components
 - Short recap - Angular CLI
 - Composing Applications with multiple modules
 - Routing and lazy loading modules
 - Loading Strategies
 - Advanced Routing
- Components
 - Content Projection
 - Smart Components/Dumb components

Agenda - 2 days- Thematic

- Day 2: **Miscellaneous**
 - Observables – pattern + design + operators
 - Angular monorepo's /micro frontends?
 - Introduction - @ngrx/store
 - Introduction - Unit Testing & E2E-testing
 - Publishing Angular libraries to NPM
 - More on Angular Schematics
 - Firebase, PWAs, Elements, ...
- Overall : Best practices on coding & architecture



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

Generic 'Advanced' Github repo

Labs, exercises and example code on the training Angular Advanced by Peter Kassenaar, info@kassenaar.com
<https://www.angulartraining.nl/>

angular training Manage topics

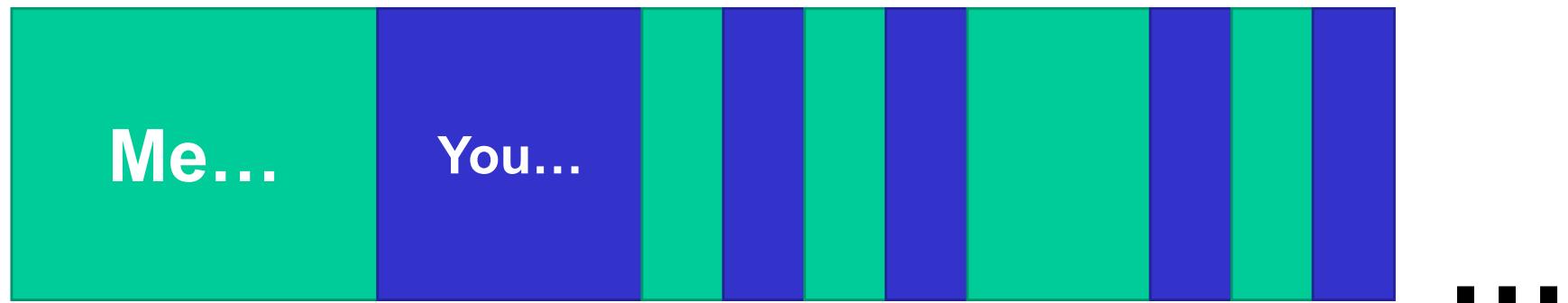
135 commits 1 branch 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Description	Last Commit
PeterKassenaar Updated examples to Angular V8	Updated examples to Angular V8	Latest commit fe74999 2 days ago
examples	Updated examples to Angular V8	2 days ago
.angulardoc.json	Updated angular.json	last year
.gitignore	Update .gitignore	2 years ago
README.md	Update README.md	27 days ago
README.md		(edit)

<https://github.com/PeterKassenaar/AngularAdvanced>

Overall process



Dia 27

PK1

Peter Kassenaar; 31-3-2020

Questions?

Dia 28

PK2

Peter Kassenaar; 31-3-2020

≡ **ANGULAR** FEATURES DOCS RESOURCES EVENTS BLOG

Search 

- Introduction
- Getting Started >
- Understanding Angular >
- Developer Guides >
- Best Practices >
- Angular Tools >
- Tutorials >
- Release Information >
- Reference > 

 - Conceptual Reference >
 - CLI Command Reference > 

 - Overview
 - Usage Analytics
 - ng add
 - ng analytics
 - ng build

CLI Overview and Command Reference

The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications directly from a command shell.

Installing Angular CLI

Major versions of Angular CLI follow the supported major version of Angular, but minor versions can be released separately.

Install the CLI using the `npm` package manager:

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

For details about changes between versions, and information about updating from previous releases, see the Releases tab on GitHub: <https://github.com/angular/angular-cli/releases>

- CLI Overview and Command Reference
- Installing Angular CLI
- Basic workflow
- Workspaces and project files
- Workspace and project configuration
- CLI command-language syntax
- Boolean options
- Relative paths
- Schematics
- Command Overview

<https://angular.io/cli>

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
 - ...

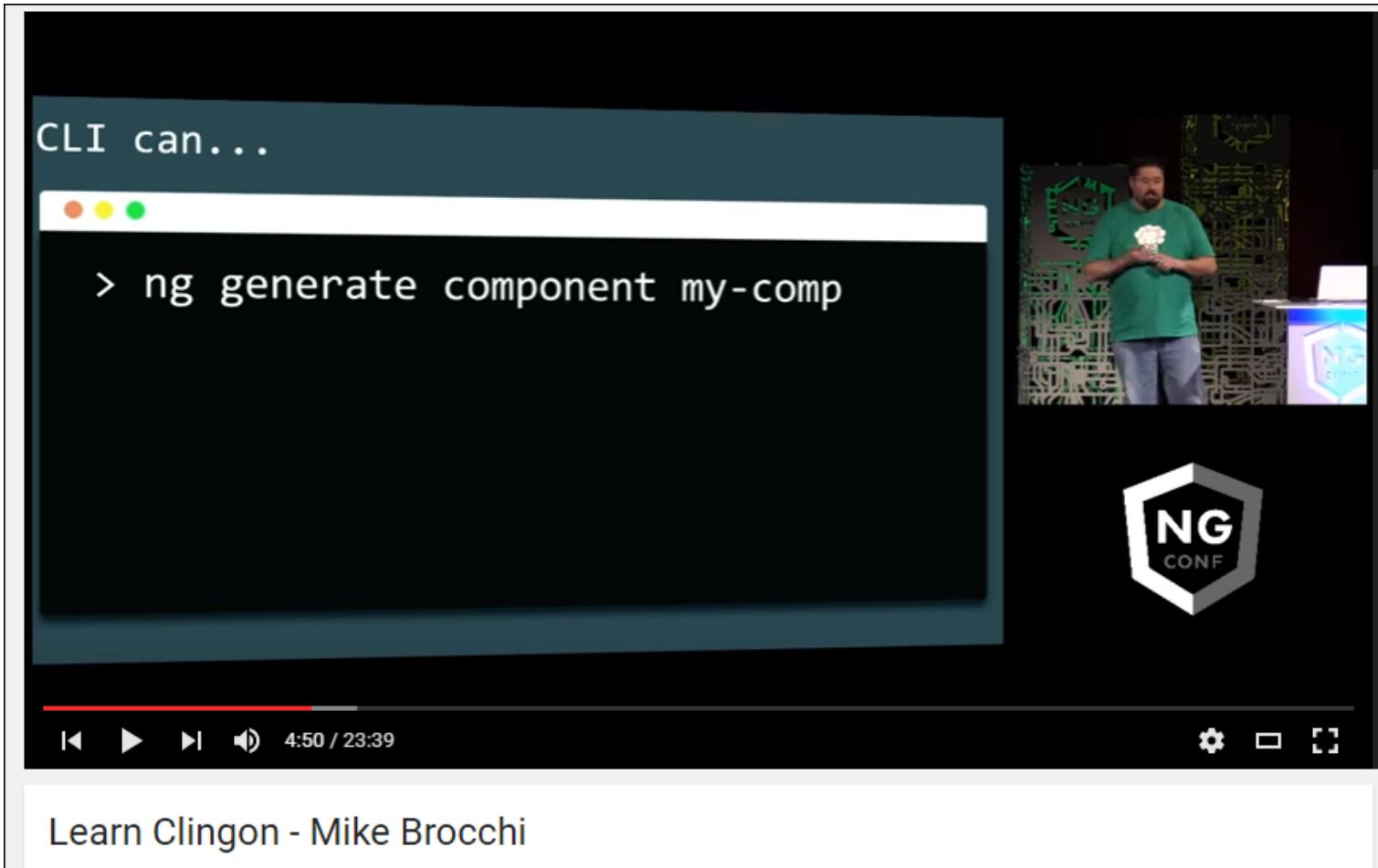
<https://cli.angular.io>

Install globally

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

CLI is also installed **locally** in project

```
npm start [--options]
```



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

Main commands

ng new – create basic app

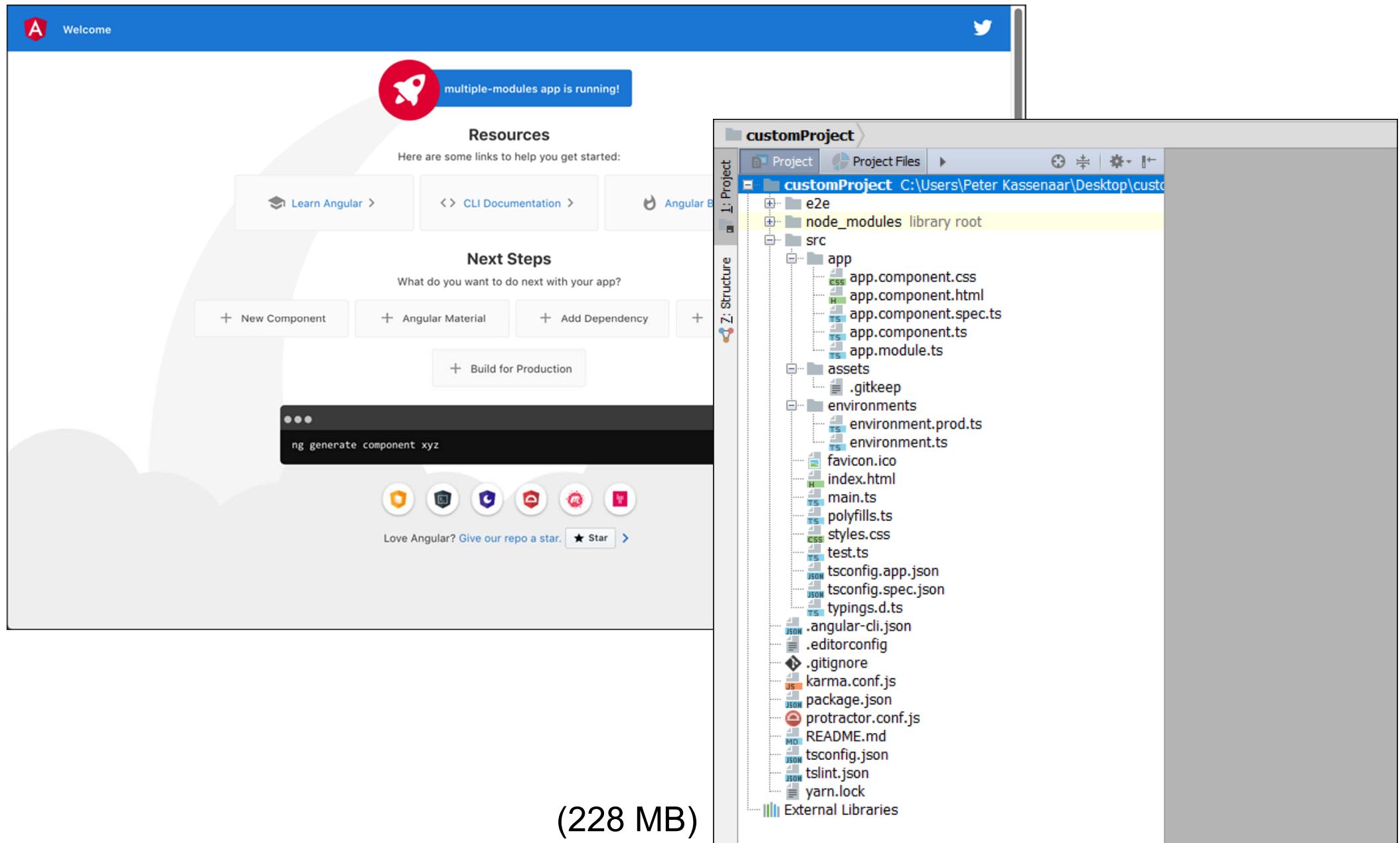
```
ng new PROJECT_NAME
```

```
cd PROJECT_NAME
```

```
ng serve // global command, npm start for local command
```

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 Command Overview page. A red box highlights the list of commands on the left side of the main content area.

Command Overview

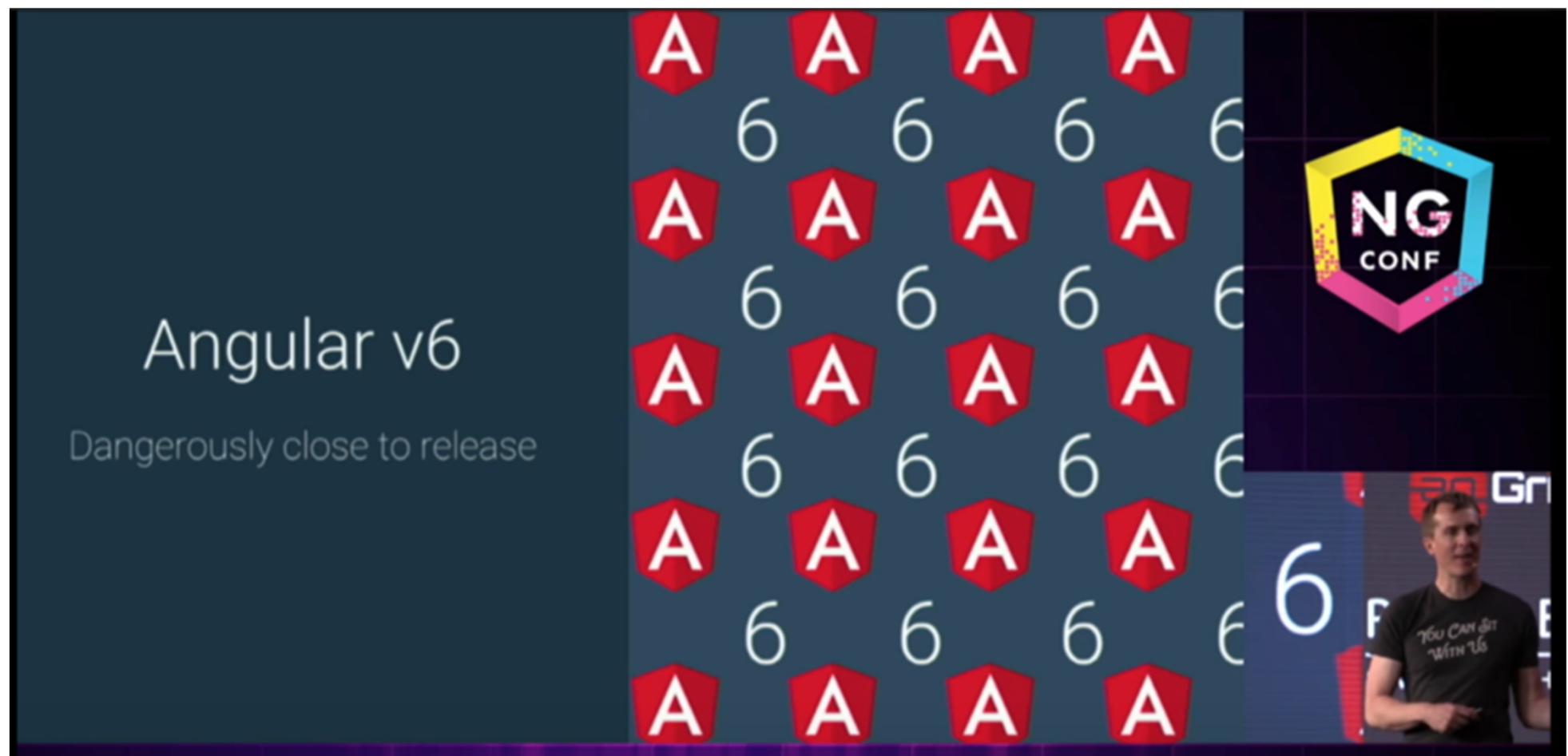
COMMAND	ALIAS	DESCRIPTION
add		Adds support for an external library to your project.
analytics		Configures the gathering of Angular CLI usage metrics. See https://v8.angular.io/cli/usage-analytics-gathering .
build	b	Compiles an Angular app into an output directory named dist/ at the given output path. 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.

CLI Overview and Command Reference

- Installing Angular CLI
- Basic workflow
- Workspaces and project files
- Workspace and project configuration
- CLI command-language syntax
- Boolean and enumerated options
- Relative paths
- Schematics
- Building with Bazel

● Command Overview

Angular CLI 6.0 - As of May 2018



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

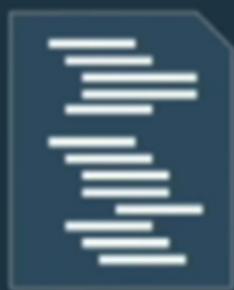
About version numbering

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

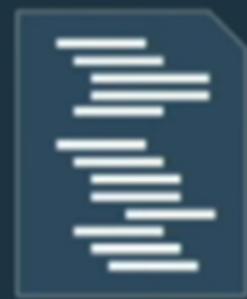
component
directive

pipe

service

...

Extending the CLI with Schematics



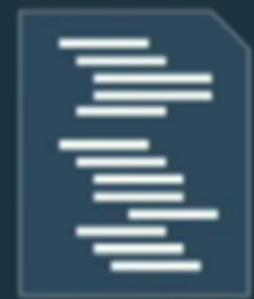
new



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 title "NG ADD" is displayed with the subtitle "Supercharge Your Angular Project". A "Submit Request!" button is visible. Below this, four add-on options are listed in cards:

- 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 with a growing list of components. Command: `ng add @angular/material`.
- Bootstrap Framework** (@NG-Bootstrap) - An Angular components collection built from the ground up using Bootstrap 4. Command: `ng add @ng-bootstrap/ng-bootstrap`.

<https://ngadd.com/>

If you want to read more - background information

The screenshot shows a web browser displaying an article on the SitePoint website. The article is titled "The Ultimate Angular CLI Reference Guide". The header includes navigation links for Courses, Books, Community, Login, and Create Free Account. Below the header, there are categories for HTML & CSS, JAVASCRIPT, PHP, DESIGN & UX, WEB, WORDPRESS, SEO BY WOORANK, and MICROSOFT TECH. The main content area features a yellow banner with wrenches and gears, indicating a technical theme. The article is categorized under JavaScript and is marked as an Article. It was written by Jurgen Van de Moere on April 25, 2017. A Microsoft advertisement is visible on the left side of the page. On the right, there is a "RECOMMENDED" sidebar with four numbered links:

- 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/>

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.

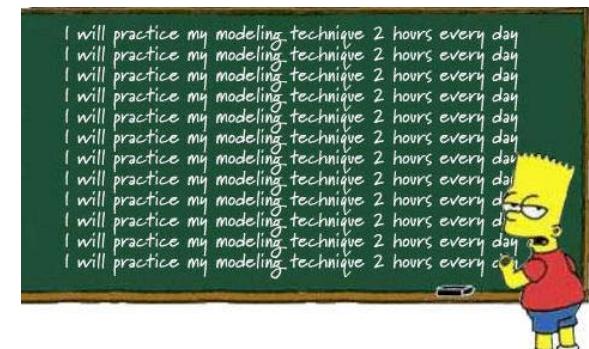
You can find the code from the examples below in [this GitHub repository](#).

Mini Workshop

- Generate a new, blank project with Angular CLI
- Generate a new component or a new service with it
- Add some new CLI packages, using `ng add` for instance
 - `@angular/material`
 - `@angular/elements`
 - See how/where they are installed
 - What files are affected?
- Study some additional CLI/ `ng` commands at <https://angular.io/cli>

For instance:

- What can you generate with `ng generate`?
- What options are available to `ng build`?
- What other commands do seem useful to you?

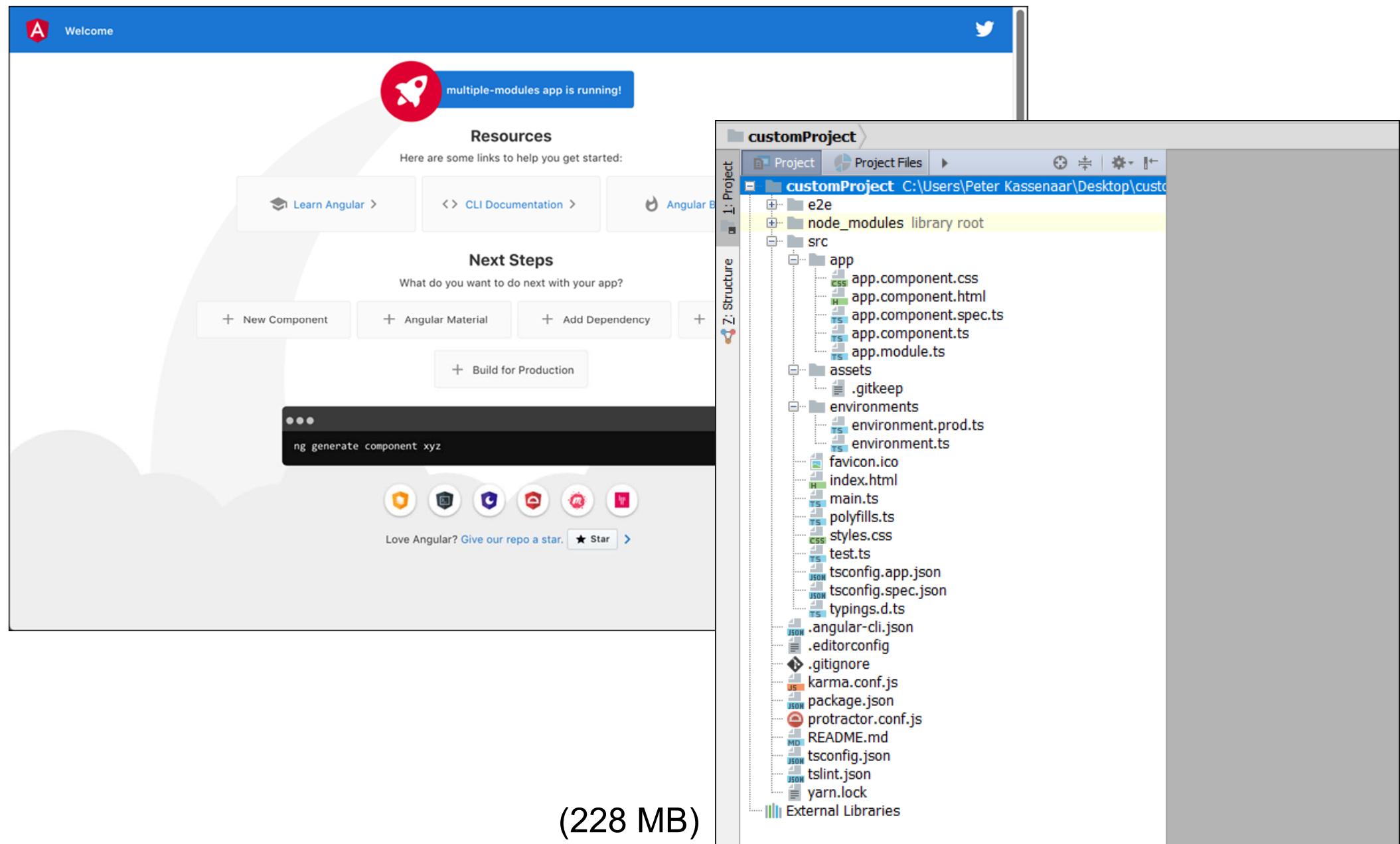




Multiple modules

Splitting your application into separate, reusable modules

Default application – 1 module

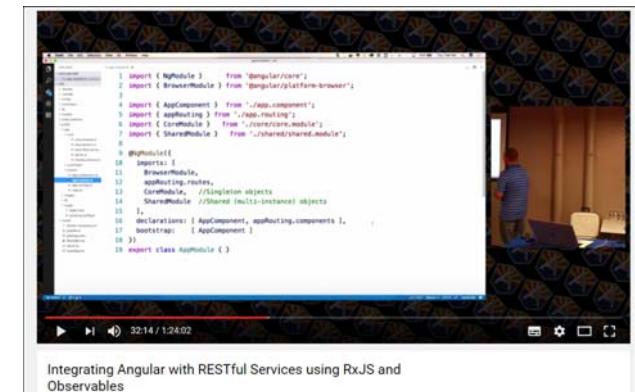


Bigger applications – multiple modules

The image shows a composite view of an Angular application. On the left, a browser window displays the Angulartraining.nl website. The site features a large red hexagonal logo with a white 'A'. Below it, the text "World-class Angular training in Dutch and English" and "Live classrooms - focused on today's developers" is visible. Two buttons, "LEARN MORE" and "SIGN UP!", are at the bottom. On the right, a file explorer window shows the project structure for "angulartraining.nl". The structure includes a "src" folder containing an "app" folder with sub-modules like "admin", "contact", "core", "dates", "home", "info", "layout", "shared", and "training", along with files such as "app.component.html", "app.component.ts", "app.module.ts", and "app-routing.module.ts". Other files in "src" include "assets", "environments", "favicon.ico", "index.html", "main.ts", "polyfills.ts", "styles.css", "test.ts", and configuration files like ".firebaserc", ".gitignore", "tsconfig.app.json", "tsconfig.spec.json", and "typings.d.ts". The file explorer also shows "environments", "favicon.ico", "index.html", "main.ts", "polyfills.ts", "styles.css", "test.ts", and configuration files like ".firebaserc", ".gitignore", "tsconfig.app.json", "tsconfig.spec.json", and "typings.d.ts".

Angular Modules

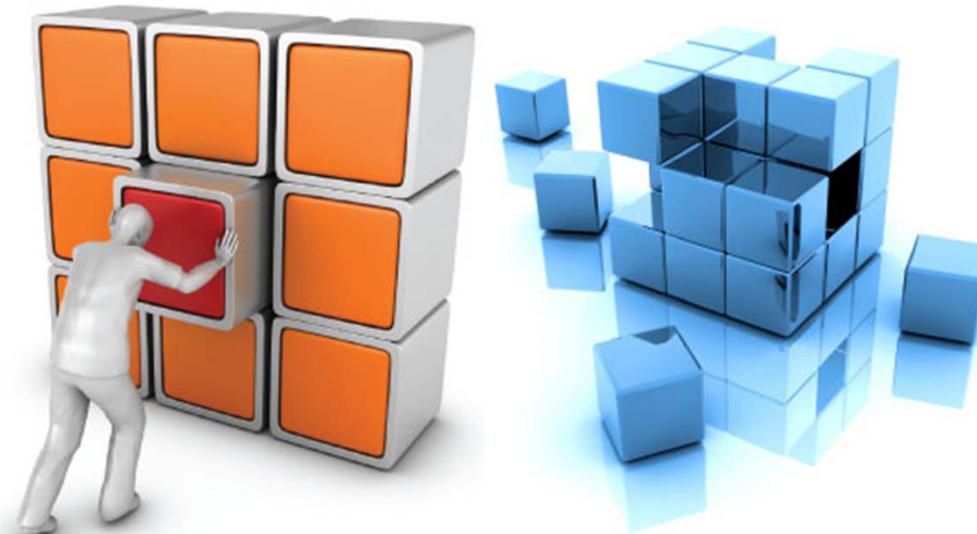
- Divide your app into *logical* and often *reusable* pieces of code
- Keyword : **code organization**
- Use one AppModule - the root of your app
- Use one CoreModule - containing all *singletons* in your app
- Use one SharedModule - containing all shared resources, possible multiple instances
- Use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

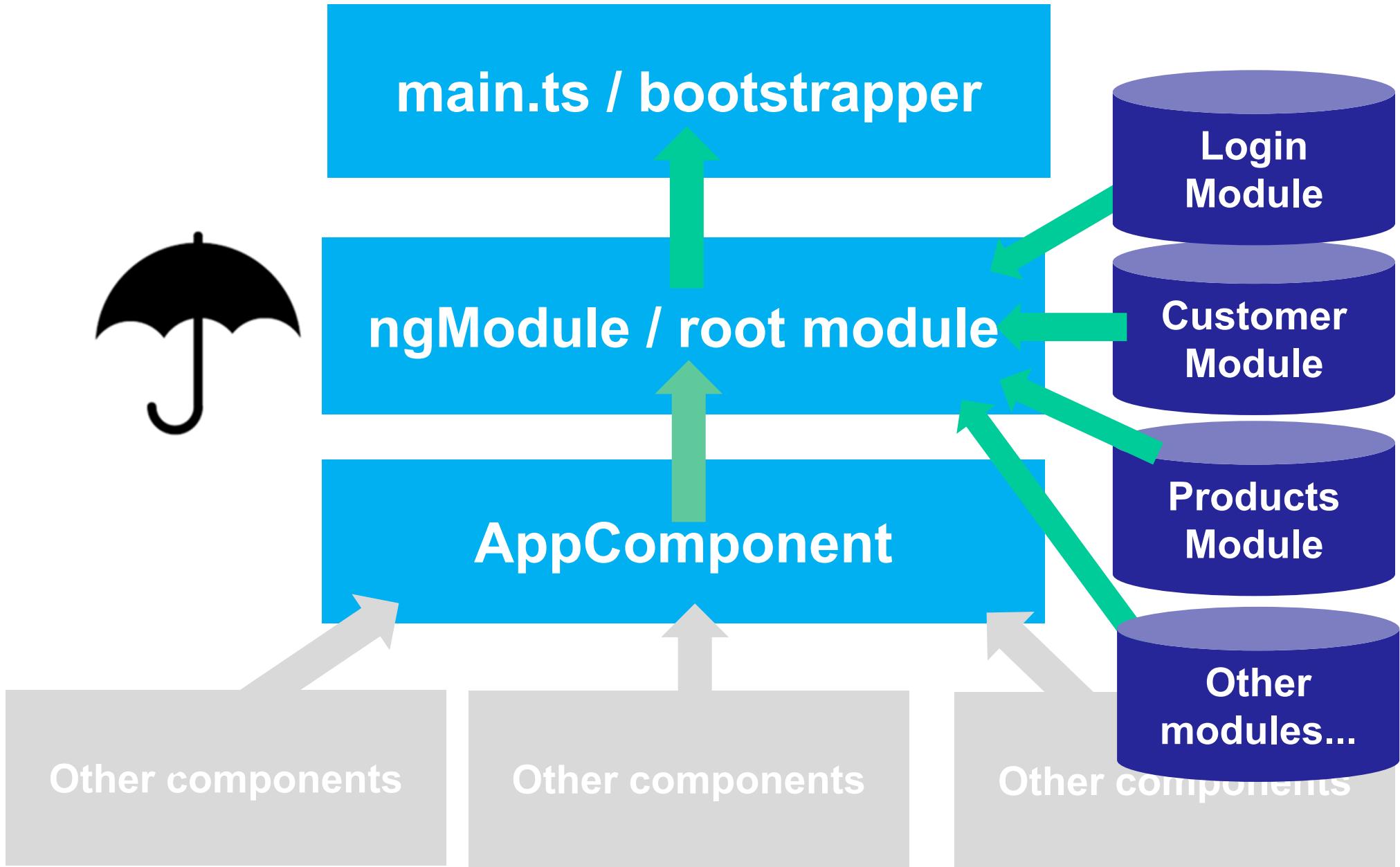


Integrating Angular with RESTful Services using RxJS and Observables

Application – multiple Modules – why?

- *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 `customers.module.ts`

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

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

- `imports: [CustomersModule]`

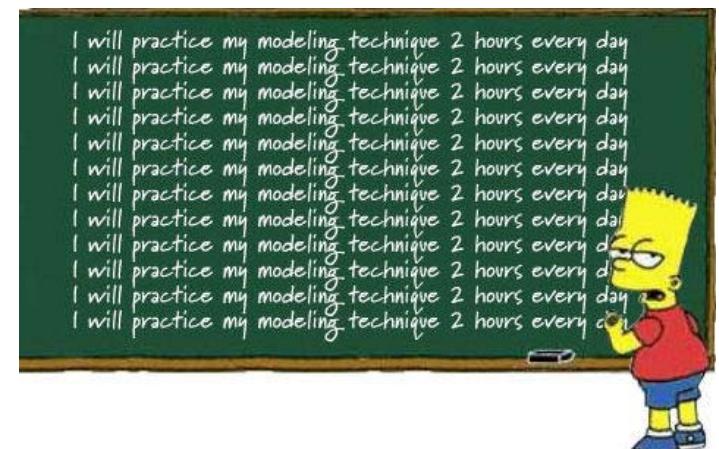
Optional : SharedModule

- 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



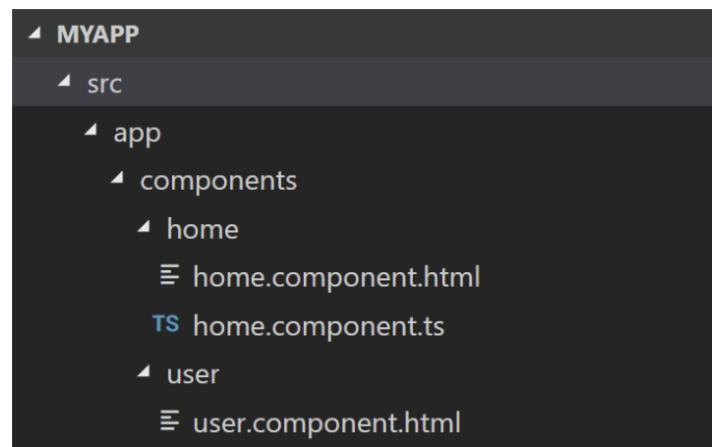
Why and how to structure Features in Modules in Angular

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:



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