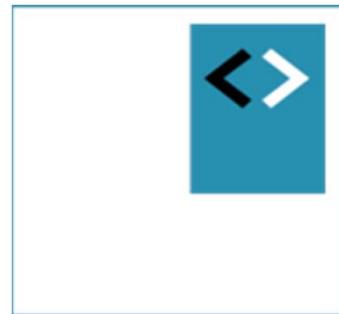


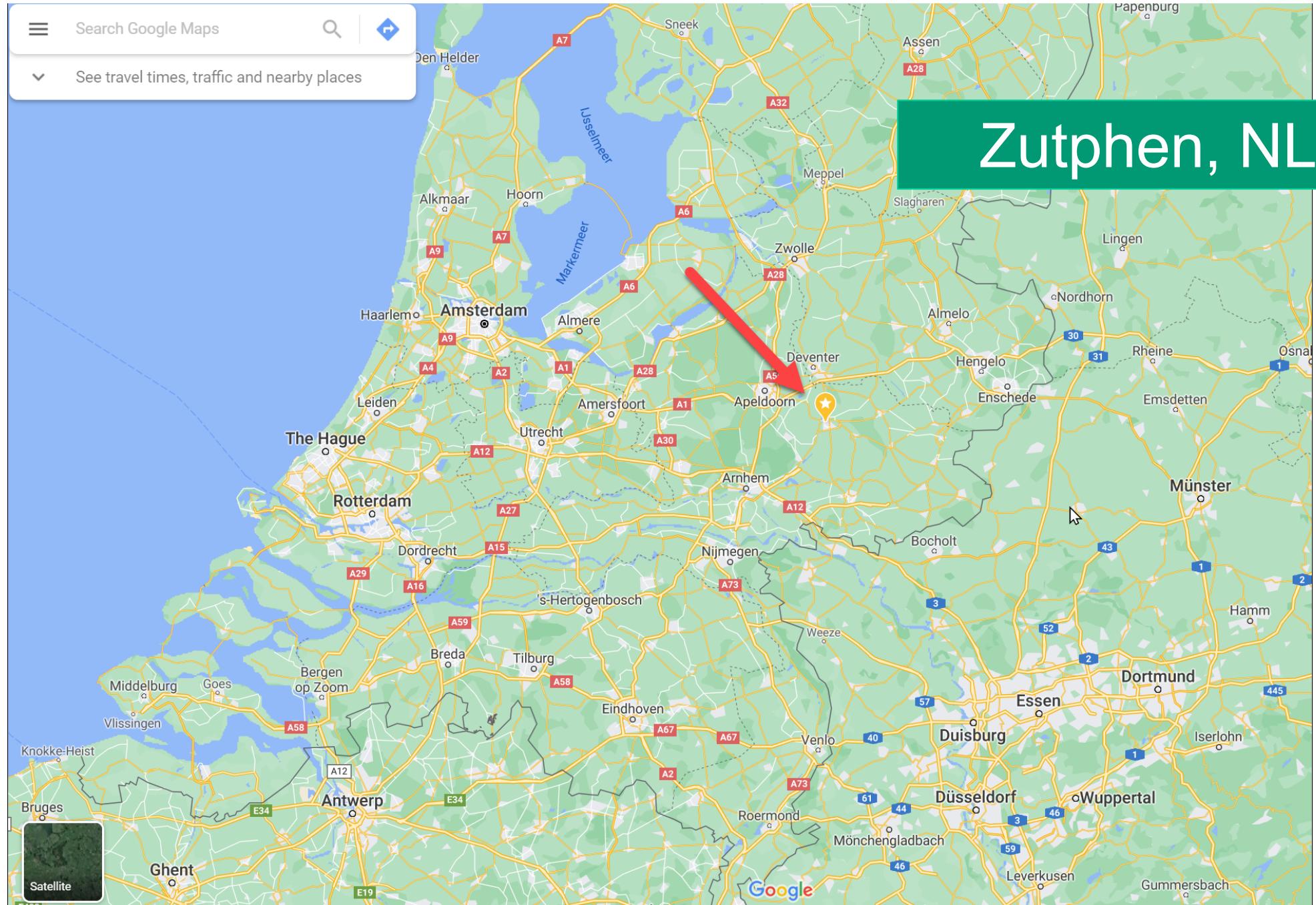


Rabobank

Angular Advanced Introduction, Architecture



Peter Kassenaar
info@kassenaar.com



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/rabo

The screenshot shows the GitHub repository page for `PeterKassenaar / rabo`. The repository is public and contains 1 branch and 0 tags. The main file listed is `README.md`, which has been updated by `PeterKassenaar`. The commit history shows two commits: an initial commit for `.gitignore` and `LICENSE`, and an update for `README.md`. The repository description states: "Slides and example code on the training Angular Advanced, spring 2023". The repository has 0 stars, 1 watching, and 0 forks. There are sections for Releases (no releases published) and Packages (no packages published). The footer includes links to GitHub's Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About pages.

PeterKassenaar / rabo Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

PeterKassenaar Update README.md a82bae5 now 2 commits

.gitignore Initial commit 11 minutes ago

LICENSE Initial commit 11 minutes ago

README.md Update README.md now

About

Slides and example code on the training Angular Advanced, spring 2023

Readme MIT license 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

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 – 6, 7, 8 February 2023

~09:00 start – Morning session

~ 10:00, 11:00 Short Break

~12:00 Lunch

~12:45 Afternoon session

~ 14:00, 15:00 Break

~16:00-16:15 - end

Wednesday: if possible,
wrap up a bit early



Material

Software (Angular + Editor + Browser + libraries)

Handouts (PDF, Github)

Workshops (in the presentations)

Websites (online)



angular.io/

Short recap

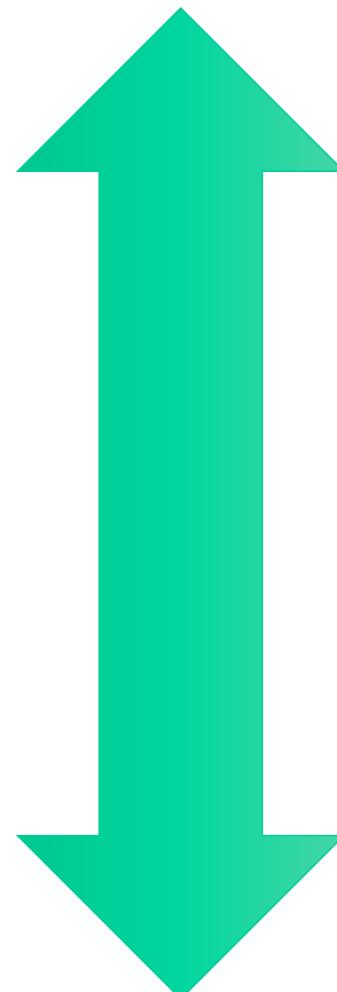
- Assumed familiar: **Fundamentals**
 - Concepts, context & architecture
 - Angular CLI basics
 - Components, Data binding
 - Services
 - Live API's
 - Component communication / event buses
 - Routing
 - Basics
 - Routing Parameters

“Advanced” Broadening?



or...

deepening?



Why devs and enterprises like Angular...



CLI



Router



HTTP



Forms



Animations



i18n



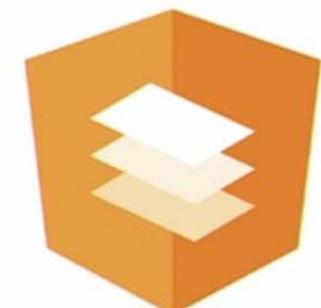
Testing



Language
Services



Universal



Material &
CDK

Agenda - 3 days - Thematic



- Day 1: **Architecture**
 - Angular CLI (tips & tricks)
 - Composing Applications with multiple modules
 - Routing and lazy loading modules
 - Loading Strategies
 - Advanced components: Smart/View ~, Dynamic ~
- Day 2: **Store & Observables**
 - Creating observables from scratch using RxJS, (un)subscribing, operators
 - Introduction - @ngrx/store
 - Concepts, State, Action, Reducer, Dispatcher, Effect, Http

Agenda - 3 days- Thematic

- Day 3: Miscellaneous
 - Unit testing
 - Dynamic Forms
 - Angular monorepo's / micro frontends?
 - Publishing Angular libraries to NPM
 - More on Angular Schematics
 - Your turn : Q & A, specific issues
 - ...
- Overall : Best practices on coding & architecture



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

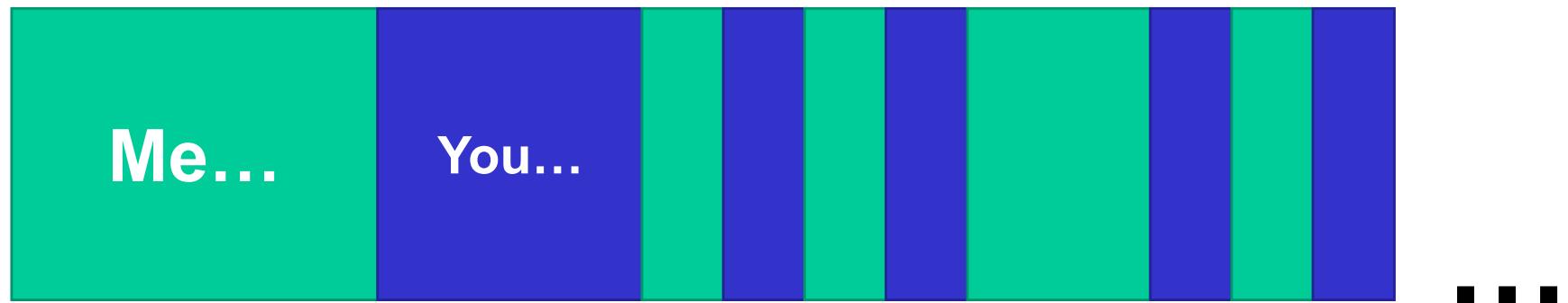
135 commits 1 branch 0 releases 2 contributors

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

File	Description	Time
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



Questions?



Angular CLI

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

≡ **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

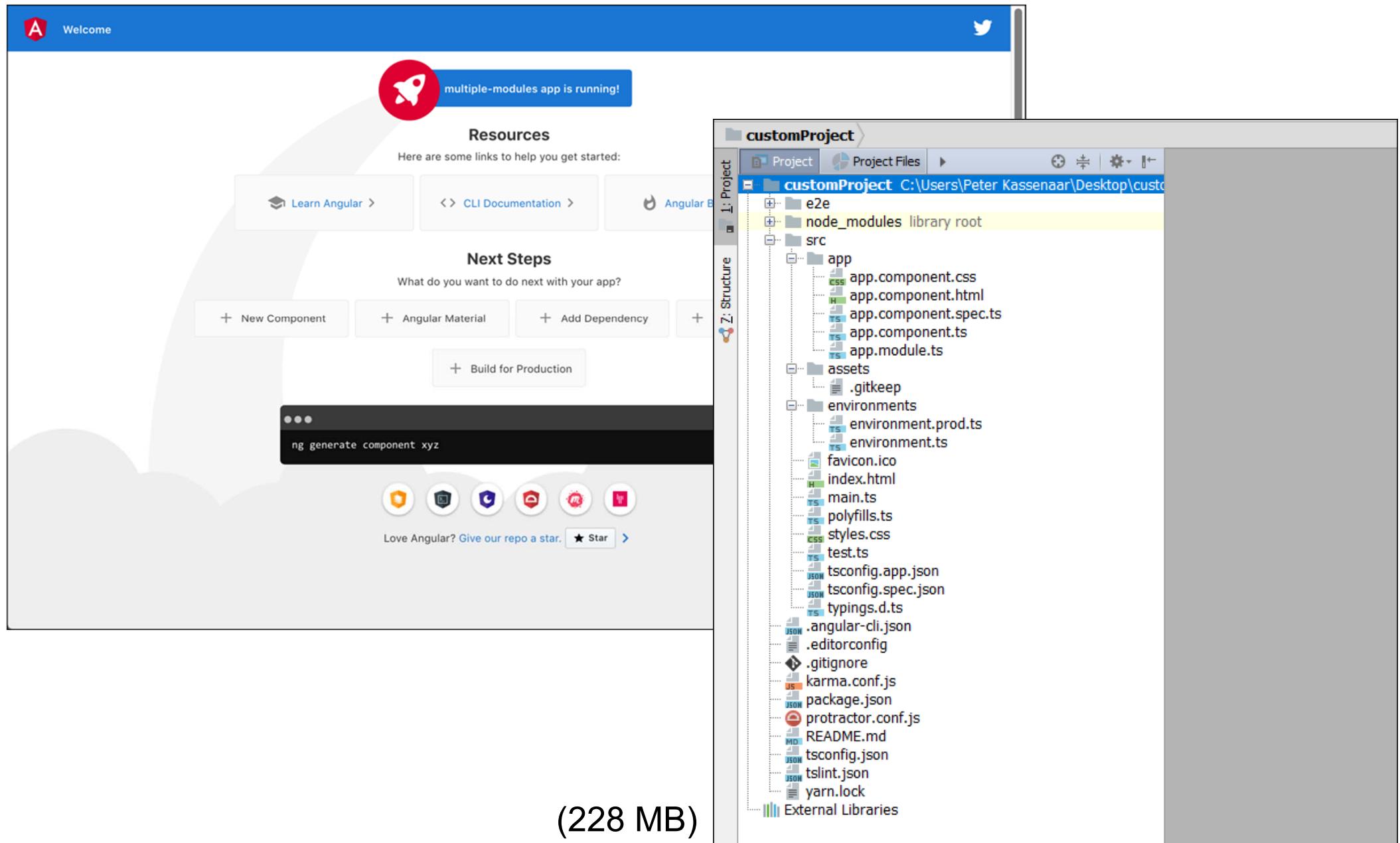
```
ng new PROJECT_NAME
```

```
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

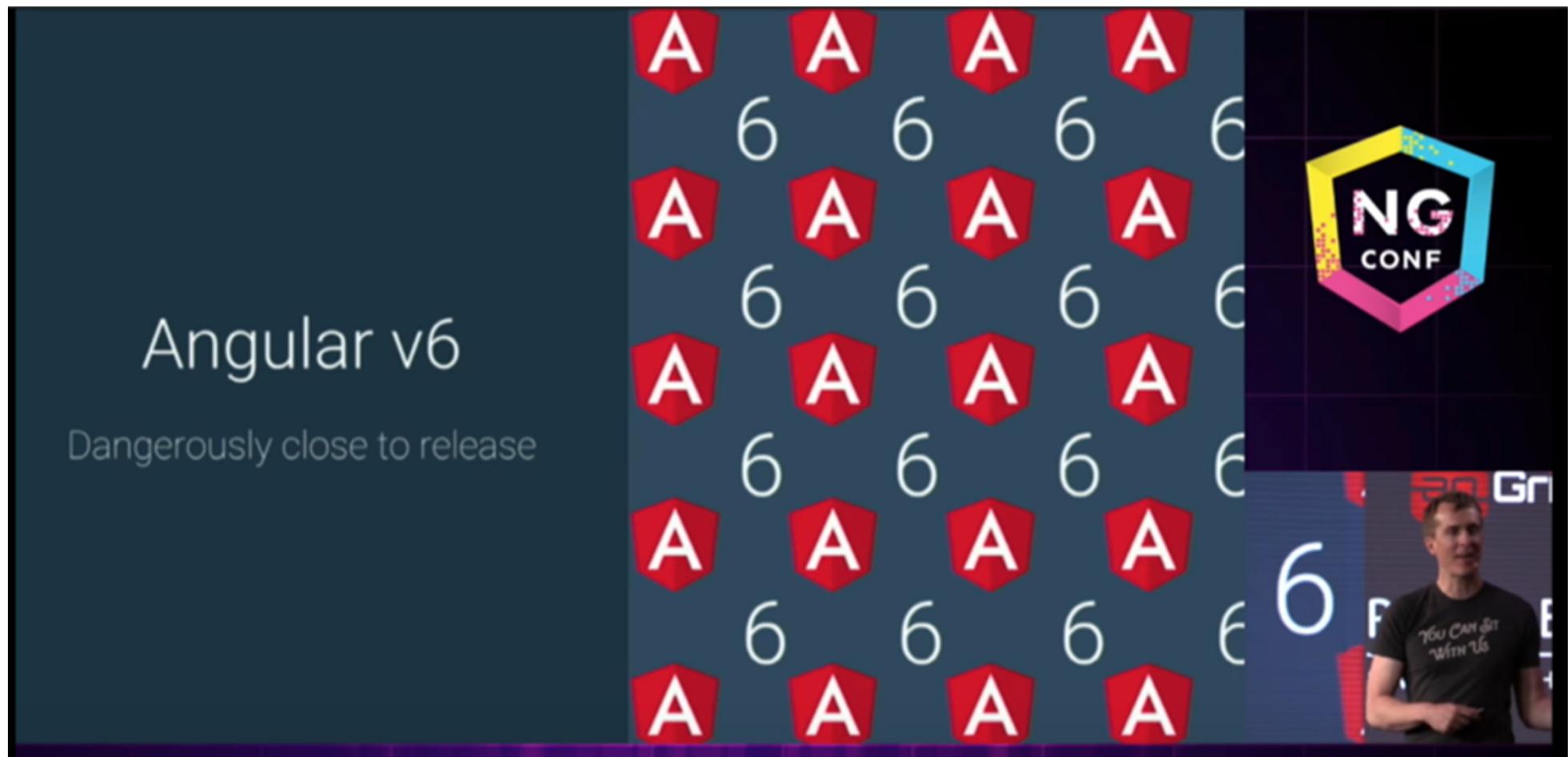
- `ng generate <blueprint> --dry-run` Do not write output files
- `ng generate <blueprint> --skip-tests` Do not write spec.ts file
- `ng generate module <name> --routing` add routing to new module

Lots (!) of options

The screenshot shows the Angular CLI Command Overview page. At the top, there's a navigation bar with links for Configuration, Release Information, Quick Reference, and a search bar. Below the navigation is a sidebar with sections like Overview, Usage Analytics, and a long list of ng commands. A red box highlights this list of commands. To the right is the main content area titled "Command Overview" which contains a table of commands with their descriptions. Further right is a sidebar with links to other CLI documentation.

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.

Angular CLI – modern version (May 2018)



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

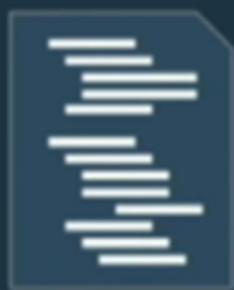
About version numbering

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

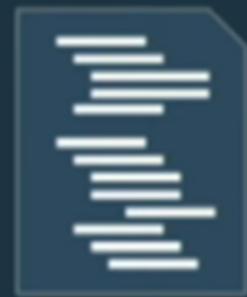


generate

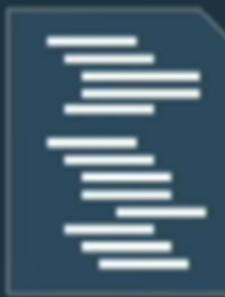
component
directive
pipe
service

...

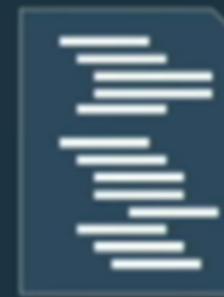
Extending the CLI with Schematics



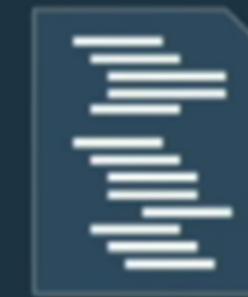
new



generate



update



add

component
directive
pipe
service

...

Info on the Angular CLI 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". Below this are two buttons: "New Library?" and "Submit Request!".

The main content area features four 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 beautiful 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://ng-add.web.app/>

If you want to learn more - background information

Save 55% On Annual Plans! 700+ Assessments, Books and Courses! **07:59:57** Get Now X

 sitepoint Join Premium

Python Programming Mobile WordPress PHP Web Entrepreneur HTML & CSS Design & UX Computing Java > Search icon

[JavaScript](#)

The Ultimate Angular CLI Reference Guide

[AngularJS](#) [Tools & Libraries](#)

 **Jurgen Van de Moere**
March 21, 2018

Share     

In this article, we'll have a look at what Angular CLI is, what it can do for you, and how it performs some of its magic behind the scenes. Even if you already use Angular CLI, this article can serve as a reference to better understand its inner workings.

Technically, you're not required to use Angular CLI to develop an Angular application, but its many features can highly improve the quality of your code and save you a lot of time along the way.

[Privacy](#)

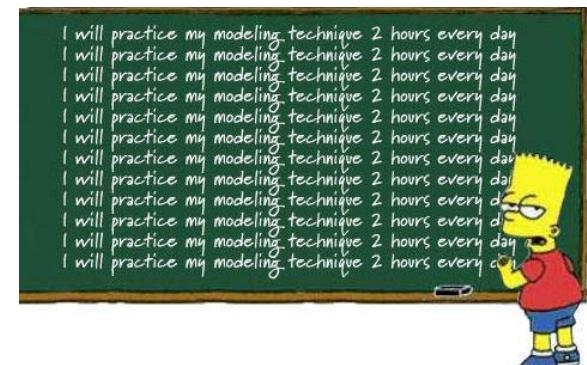
<https://www.sitepoint.com/ultimate-angular-cli-reference/>

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`
 - `ngx-bootstrap` (<https://valor-software.com/ngx-bootstrap/#/>)
 - See how/where they are installed
 - What files are affected?
 - <https://ng-add.web.app/>
- 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

The image shows the Angular CLI welcome screen on the left and a file explorer on the right.

Angular CLI Welcome Screen:

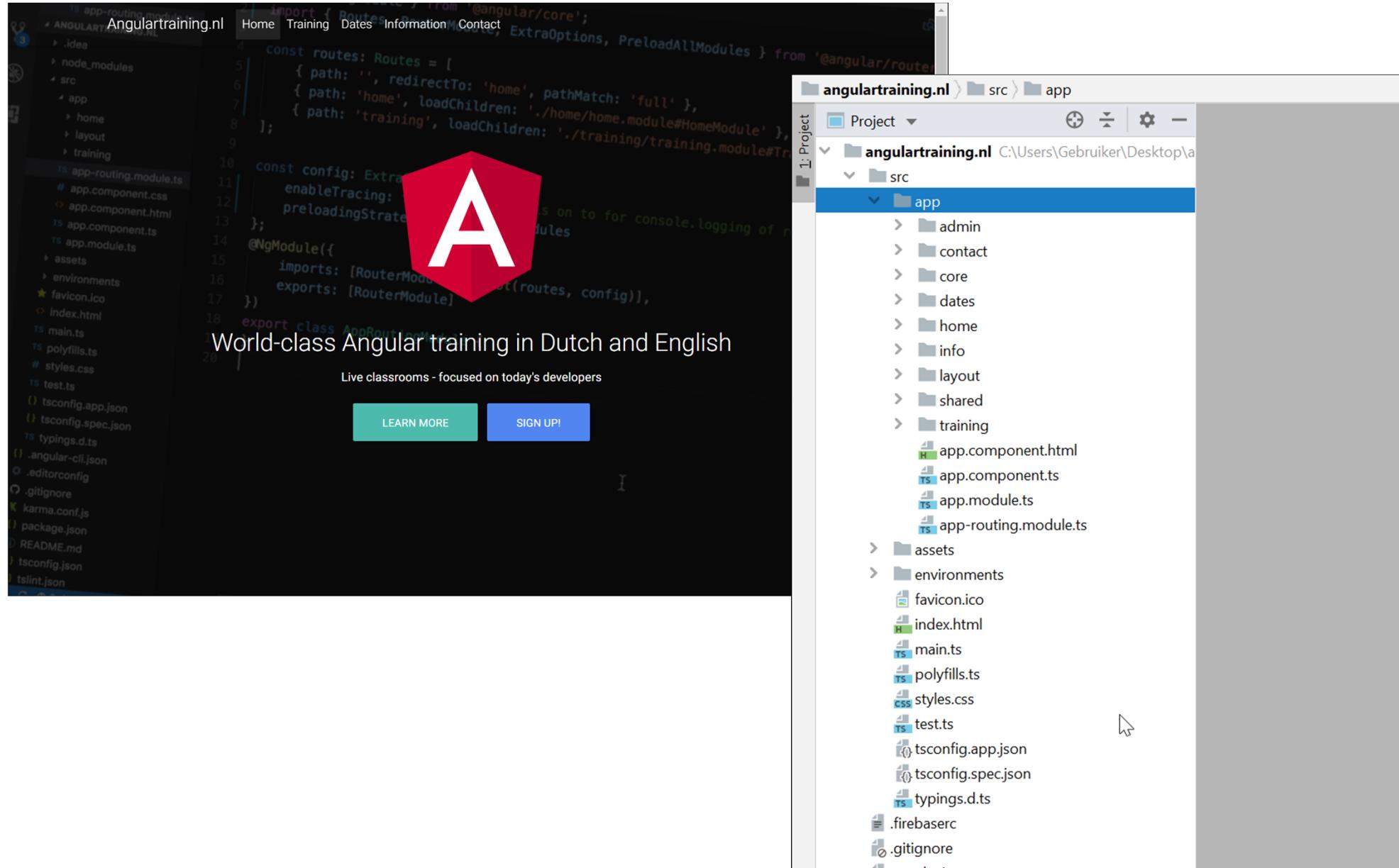
- Header:** Welcome, Twitter icon.
- Notification:** multiple-modules app is running!
- Resources:** Links to Learn Angular, CLI Documentation, and Angular Build.
- Next Steps:** Buttons for New Component, Angular Material, Add Dependency, and Build for Production.
- Terminal Mockup:** Shows the command `ng generate component xyz`.
- Icons:** Icons for various Angular tools like ng, rxjs, etc.
- Call to Action:** Love Angular? Give our repo a star.

File Explorer (customProject structure):

- Project Root:** customProject (C:\Users\Peter Kassenaar\Desktop\customProject)
- Structure:** Z: \ Project
- node_modules:** library root (highlighted in yellow)
- src:**
 - app:** app.component.css, app.component.html, app.component.spec.ts, app.component.ts, app.module.ts
 - assets:** .gitkeep
 - environments:** environment.prod.ts, environment.ts
 - favicon.ico**
 - index.html**
 - main.ts**
 - polyfills.ts**
 - styles.css**
 - test.ts**
 - tsconfig.app.json**
 - tsconfig.spec.json**
 - typings.d.ts**
 - .angular-cli.json**
 - .editorconfig**
 - .gitignore**
 - karma.conf.js**
 - package.json**
 - protractor.conf.js**
 - README.md**
 - tsconfig.json**
 - tslint.json**
 - yarn.lock**
- External Libraries:**

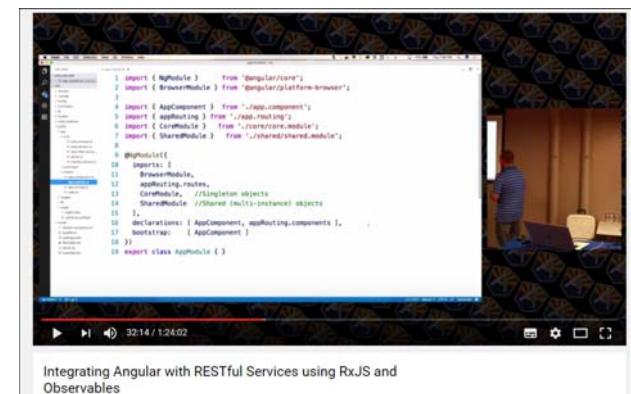
(228 MB)

Bigger applications – multiple modules



Angular Modules - naming

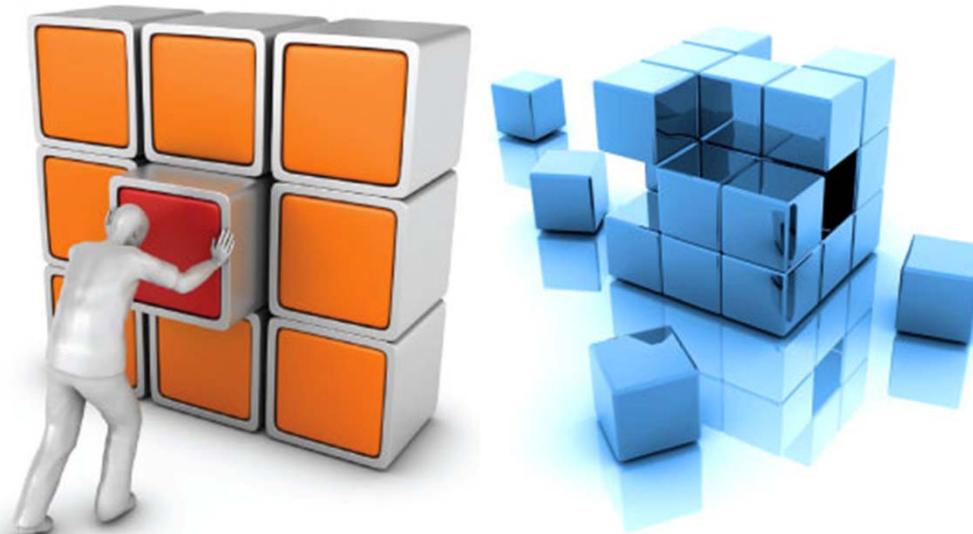
- Divide your app into *logical* and often *reusable* pieces of code
- Keyword : **code organization**
- Use one AppModule - the root of your app
- Use one CoreModule - containing all *singletons* in your app
- Use one SharedModule - containing all shared resources, possible multiple instances
- Use 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.



Since Angular 15 – Standalone Components

- Applications and components without an `NgModule`

*"Standalone components provide a **simplified way** to build Angular applications. Standalone components, directives, and pipes aim to streamline the authoring experience by reducing the need for NgModules.*

*Existing applications can **optionally and incrementally** adopt the new standalone style without any breaking changes."*

More info on standalone components

The screenshot shows a browser window displaying the Angular documentation. The URL in the address bar is `https://angular.io/guide/standalone-components`. The page title is "Getting started with standalone components". The left sidebar contains a navigation menu with items like "Introduction", "Getting started", "Understanding Angular", "Developer guides" (which is expanded to show "Overview", "Standalone components", "Change detection", "Routing and navigation", "Forms", "HTTP client", "Image optimization", "Testing", "Internationalization", "Animations", and "Service Workers & PWA"), and "Service workers & PWA". The main content area starts with a paragraph about standalone components, followed by a section titled "Creating standalone components" featuring a YouTube thumbnail for a video titled "Getting Started with Standalone Components". To the right of the main content, there is a sidebar with a list of related topics: "Getting started with standalone components", "Creating standalone components", "The standalone flag and component imports", "Using existing NgModules in a standalone component", "Using standalone components in NgModule-based applications", "Bootstrapping an application using a standalone component", "Configuring dependency injection", "Routing and lazy-loading", "Lazy loading a standalone component", and "Lazy loading many routes at once".

<https://angular.io/guide/standalone-components>

Verdict (personal, opinion!)

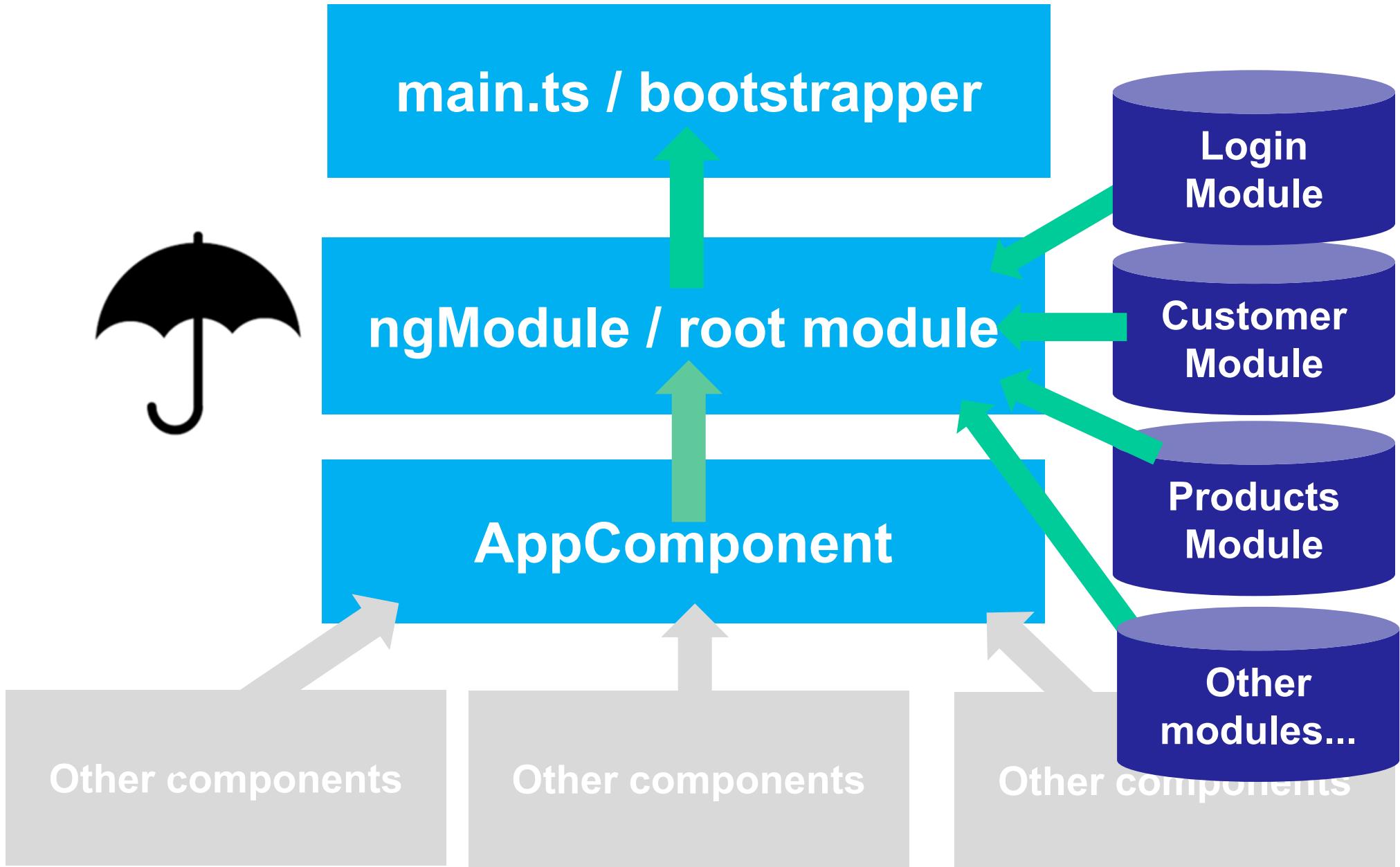
We're NOT using standalone components in this course

- Standalone components **might come in handy**, in new applications from scratch
- Due to confusion, personally I **don't see it implemented 'optionally and incrementally'** in existing apps
- It is mostly done (IMO) to **mimic Vue, React** et all.

But, if you want more information...

- <https://blog.angular.io/angular-v15-is-now-available-df7be7f2f4c8>
- <https://www.angulararchitects.io/aktuelles/angulars-future-without-ngrx-modules-lightweight-solutions-on-top-of-standalone-components/>
- <https://netbasal.com/angular-standalone-components-welcome-to-a-world-without-ngrx-module-abd3963e89c5>





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]`

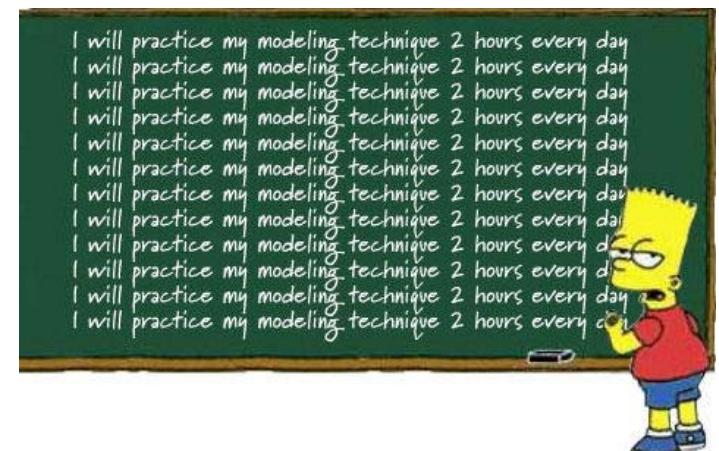
Optional : SharedModule

- 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 (npm install, npm start)
 - 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
          TS home.component.ts
        ▲ user
          └ user.component.html
```

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