

ilionx



Global Knowledge®

Angular Module – Advanced Introduction

Peter Kassenaar –
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

Peter Kassenaar

- Trainer, author, developer – sinds 1996
- Specialty: *"Everything JavaScript"*
- JavaScript, ES6, Angular, NodeJS, TypeScript, jQuery, PhoneGap, Ionic

www.kassenaar.com/blog

info@kassenaar.com

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



ING

OHRA

euricom
A DIMENSION DATA COMPANY

sanoma

delta lloyd

zenito
BETERE ZEKERHEID
VOOR ONDERNEMERS

Atos

woonbron
OBERON INTERACTIVE

ROC West-Brabant

the eforum
FACTORY

Angulartraining.nl

Home

Training

Dates

Information

Contact

2018 dates now available!



World-class Angular training in Dutch and English

Live classrooms - focused on today's developers

LEARN MORE

SIGN UP!

www.angulartraining.nl

About you...



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?

github.com/PeterKassenaar/ilionx

The screenshot shows the GitHub repository page for `PeterKassenaar / ilionx`. The repository is described as "Slides en voorbeeldcode bij de training Angular Advanced - Ilionx, voorjaar 2020". It has 7 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The latest commit is by PeterKassenaar, adding an extra link, 21 days ago. The repository contains files `slides`, `.gitignore`, and `README.md`. The `README.md` file is visible at the bottom of the screenshot, showing the repository name `ilionx` and the description.

Search or jump to... Pull requests Issues Marketplace Explore

Your repository details have been saved.

PeterKassenaar / ilionx Unwatch 1 Star 0 Fork 1

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Slides en voorbeeldcode bij de training Angular Advanced - Ilionx, voorjaar 2020 Edit

Manage topics

7 commits 1 branch 0 packages 0 releases 1 contributor

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

PeterKassenaar Added extra link Latest commit 14aa40b 21 days ago

slides	addes slides on day #3	21 days ago
.gitignore	Added link to interceptors	22 days ago
README.md	Added extra link	21 days ago

README.md

ilionx

Slides en voorbeeldcode bij de training Angular Advanced - Ilionx, voorjaar 2020

Agenda

1-2 April 2020 – wo + do

20-21 April 2020 – ma + di – *voorlopig*

~ 18:00 start

~ 19:30 Break

~ 21:00 einde

Agenda - 4 nights - Thematic

- Day 1: **Architectuur**
 - Composing Applications with multiple modules
 - Routing and lazy loading modules
 - Loading Strategies
 - Advanced components
- Day 2: **Observables**
 - Observables from scratch, more operators
 - Examples (typeahead, fetching data from multiple sources)

Agenda - 4 nights - Thematic

- Day 3: **State Management**
 - State management with @ngrx/store
 - Concepts – actions, dispatchers, reducers, store
 - Payload, complex types
- Day 4: **Enterprise applications**
 - Working with monorepo's, Angular CLI
 - More on Angular Schematics

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

The screenshot shows the GitHub repository page for **PeterKassenaar / AngularAdvanced**. The repository description is "Labs, exercises and example code on the training Angular Advanced by Peter Kassenaar, info@kassenaar.com" with a link to <https://www.angulartraining.nl/>. The repository has 135 commits, 1 branch, 0 releases, and 2 contributors. The latest commit by PeterKassenaar is "Updated examples to Angular V8" (commit fe74999) from 2 days ago. The commit history table lists the following changes:

File	Commit Message	Time
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

The repository also features a sidebar with "angular" and "training" topics, and a "Manage topics" button. The bottom of the page shows the "README.md" file selected.

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



Multiple modules

Splitting your application into separate, reusable modules

Default application – 1 module

The image displays the Angular CLI interface on the left and a file explorer on the right, illustrating the default application structure.

Angular CLI Interface:

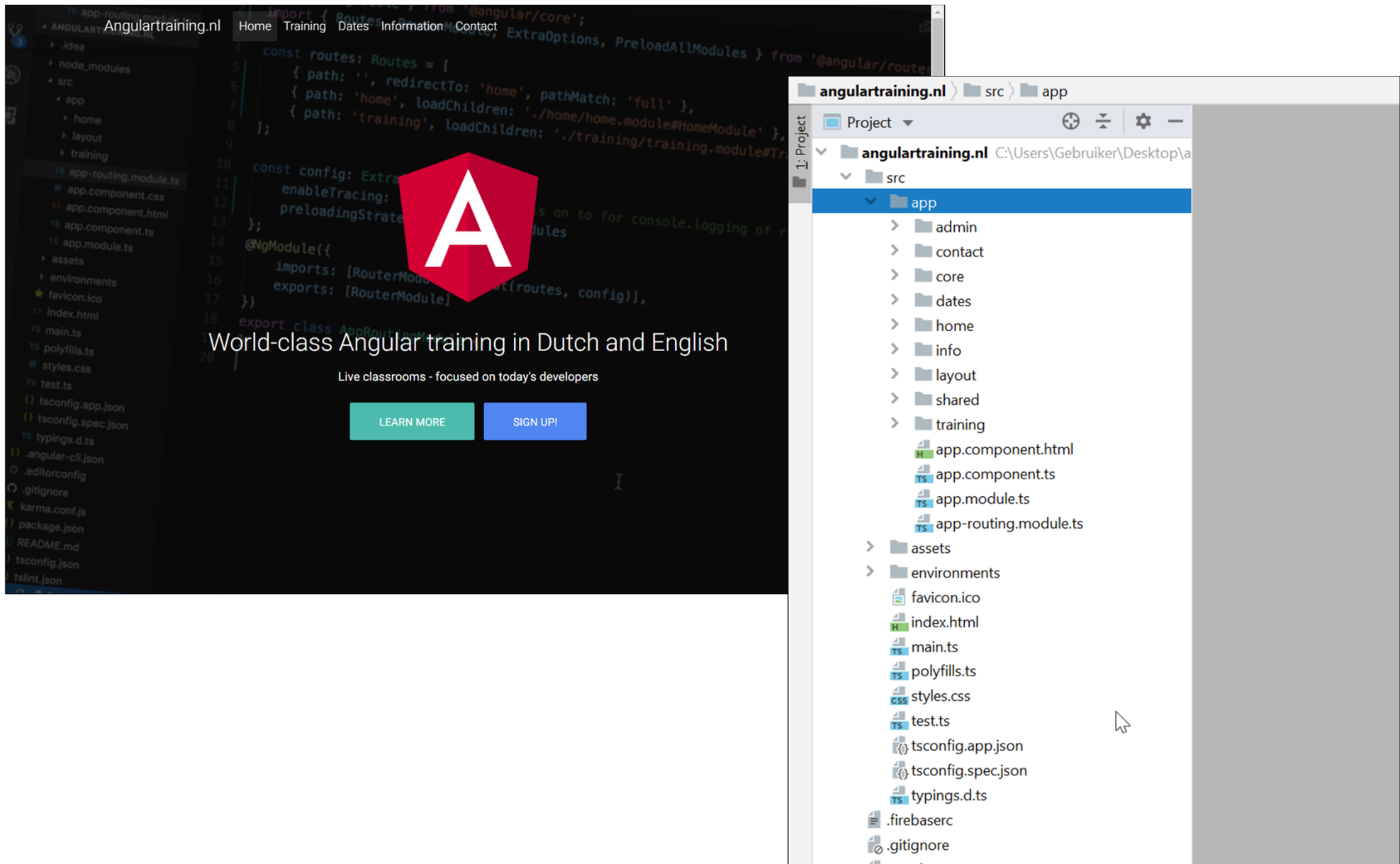
- Welcome:** The top bar shows the Angular logo and the word "Welcome".
- multiple-modules app is running!** A red notification bubble with a rocket icon.
- Resources:** A section titled "Resources" with the text "Here are some links to help you get started:". It includes links for "Learn Angular", "CLI Documentation", and "Angular B".
- Next Steps:** A section titled "Next Steps" with the text "What do you want to do next with your app?". It includes buttons for "New Component", "Angular Material", "Add Dependency", and "Build for Production".
- Terminal:** A terminal window showing the command `ng generate component xyz`.
- Footer:** A footer with social media icons and the text "Love Angular? Give our repo a star. ★ Star >".

File Explorer:

- Project:** The file explorer shows the project structure for `customProject` located at `C:\Users\Peter Kassenaar\Desktop\custo`.
- Structure:** The file explorer shows the following structure:
 - `customProject` (library root)
 - `e2e`
 - `node_modules` (library root)
 - `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



The image shows a development environment with two main windows. The left window is a code editor displaying Angular routing configuration. The right window is a file explorer showing the project structure.

Code Editor (Left):

- File: `angulartraining.nl`
- Routes configuration:

```
const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', loadChildren: './home/home.module#HomeModule' },
  { path: 'training', loadChildren: './training/training.module#TrainingModule' }
];
```
- NgModule configuration:

```
@NgModule({
  imports: [RouterModule.forRoot(routes, config)],
  exports: [RouterModule]
})
```

File Explorer (Right):

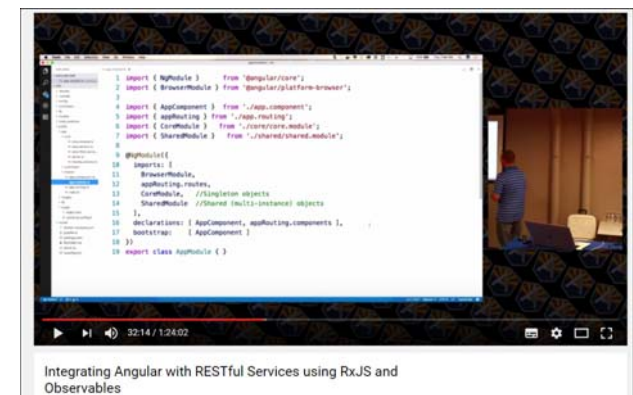
- Project: `angulartraining.nl`
- src
- app
 - admin
 - contact
 - core
 - dates
 - home
 - info
 - layout
 - shared
 - training
 - app.component.html
 - app.component.ts
 - app.module.ts
 - app-routing.module.ts
- assets
- environments
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - styles.css
 - test.ts
 - tsconfig.app.json
 - tsconfig.spec.json
 - typings.d.ts
- .firebaserc
- .gitignore

Angulartraining.nl Website Overlay:

- Navigation: Home, Training, Dates, Information, Contact
- Logo: Angular (red shield with white 'A')
- Text: "World-class Angular training in Dutch and English"
- Text: "Live classrooms - focused on today's developers"
- Buttons: [LEARN MORE](#) (green), [SIGN UP!](#) (blue)

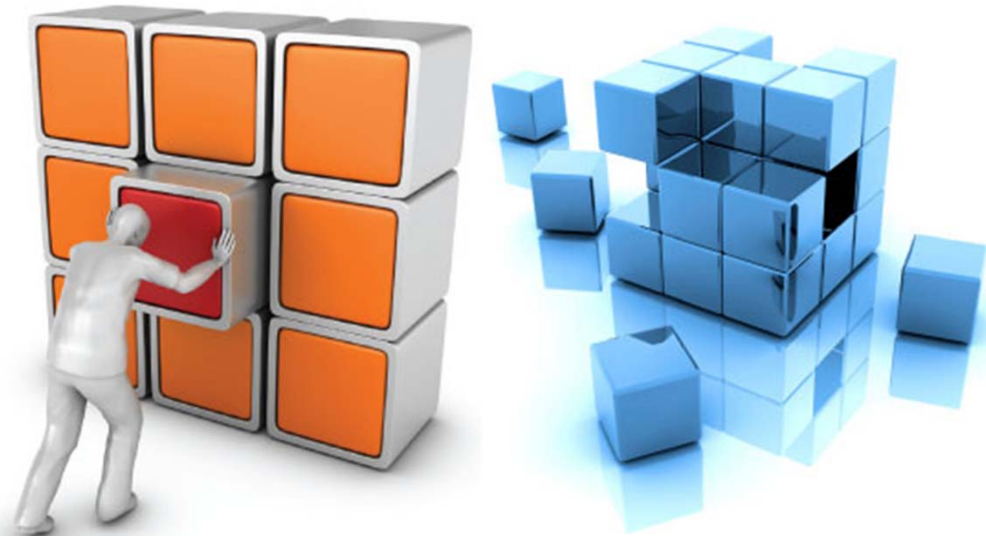
Angular Modules

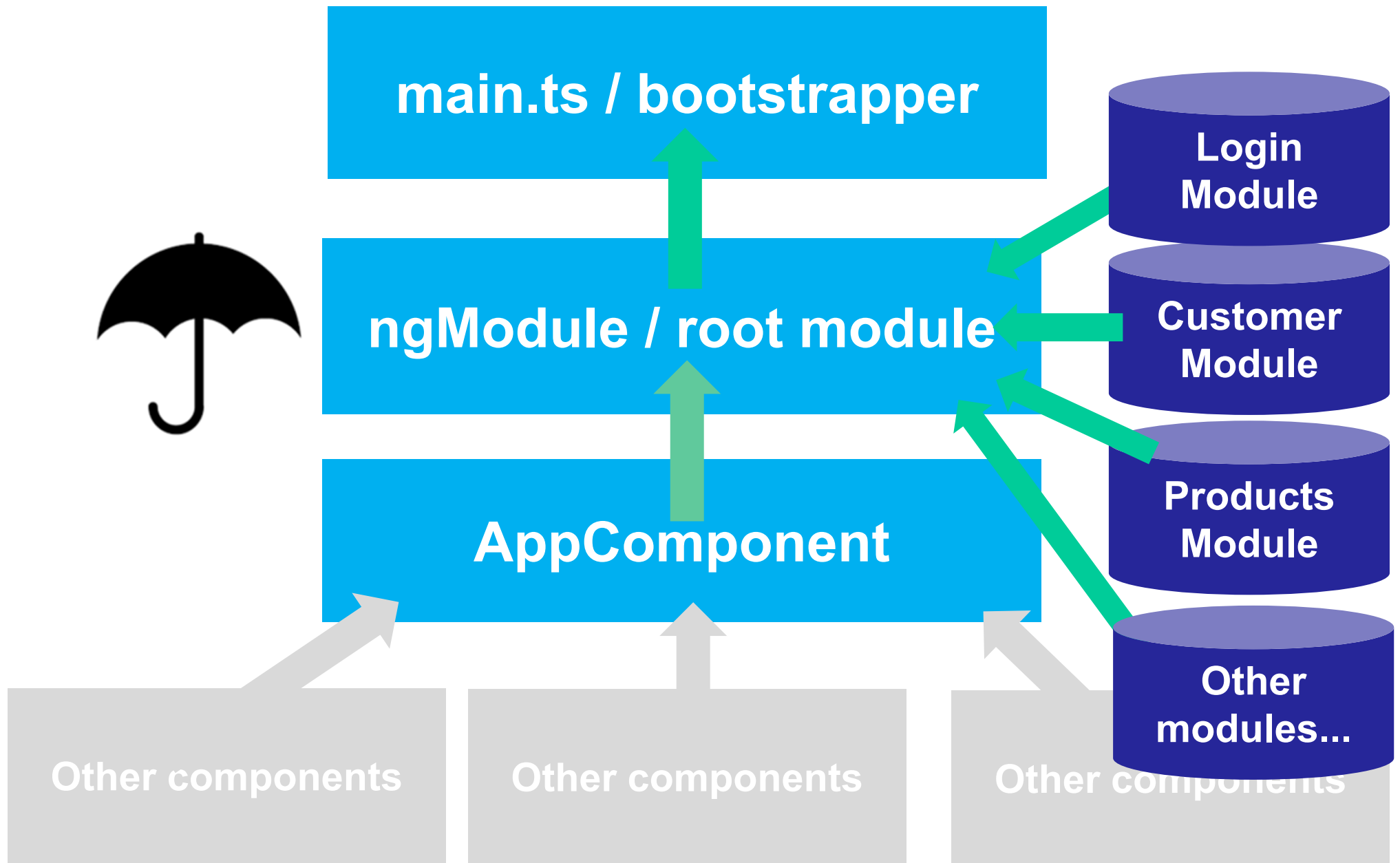
- Divide your app into *logical* and often *reusable* pieces of code
- Keyword : **code organization**
- Use one AppModule - the root of your app
- Use one CoreModule - containing all *singletons* in your app
- Use one SharedModule - containing all shared resources, possible multiple instances
- Use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



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 `customer.module.ts`

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

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

- `imports: [CustomerModule]`

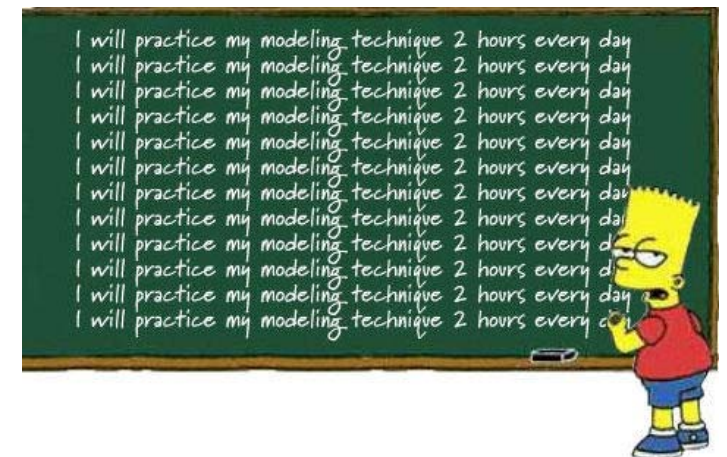
Optional : SharedModule

- 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>