Gemeente Haarlem

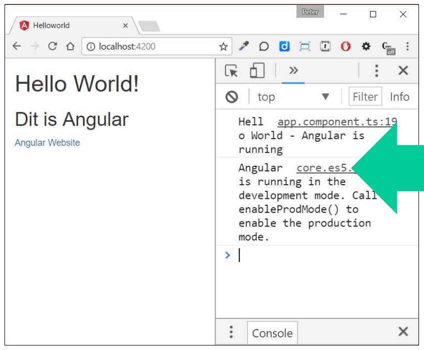
Short Recap – day #1

Peter Kassenaar
info@kassenaar.com

Agenda - day #1

- Angular New Features + refresher
    - NPM, npx, `node_modules`, webpack, Vite and more
    - Files – `package.json`, `angular.json`, `tsconfig.json`, `app.config.ts`
    - `@for`, `@if-then-else`, `@switch()`
    - Standalone components and module-less defaults
    - Dependency Injection
    - Updating Angular versions
    - Dependency Injection
    - Getters vs. Methods, `types` vs `interfaces`
    - New `provideHttpClient()`
    - Subscribing vs. Async pipe
- Routing & Guards
    - Creating & using guards, lazy loading

Structure: modern Angular apps

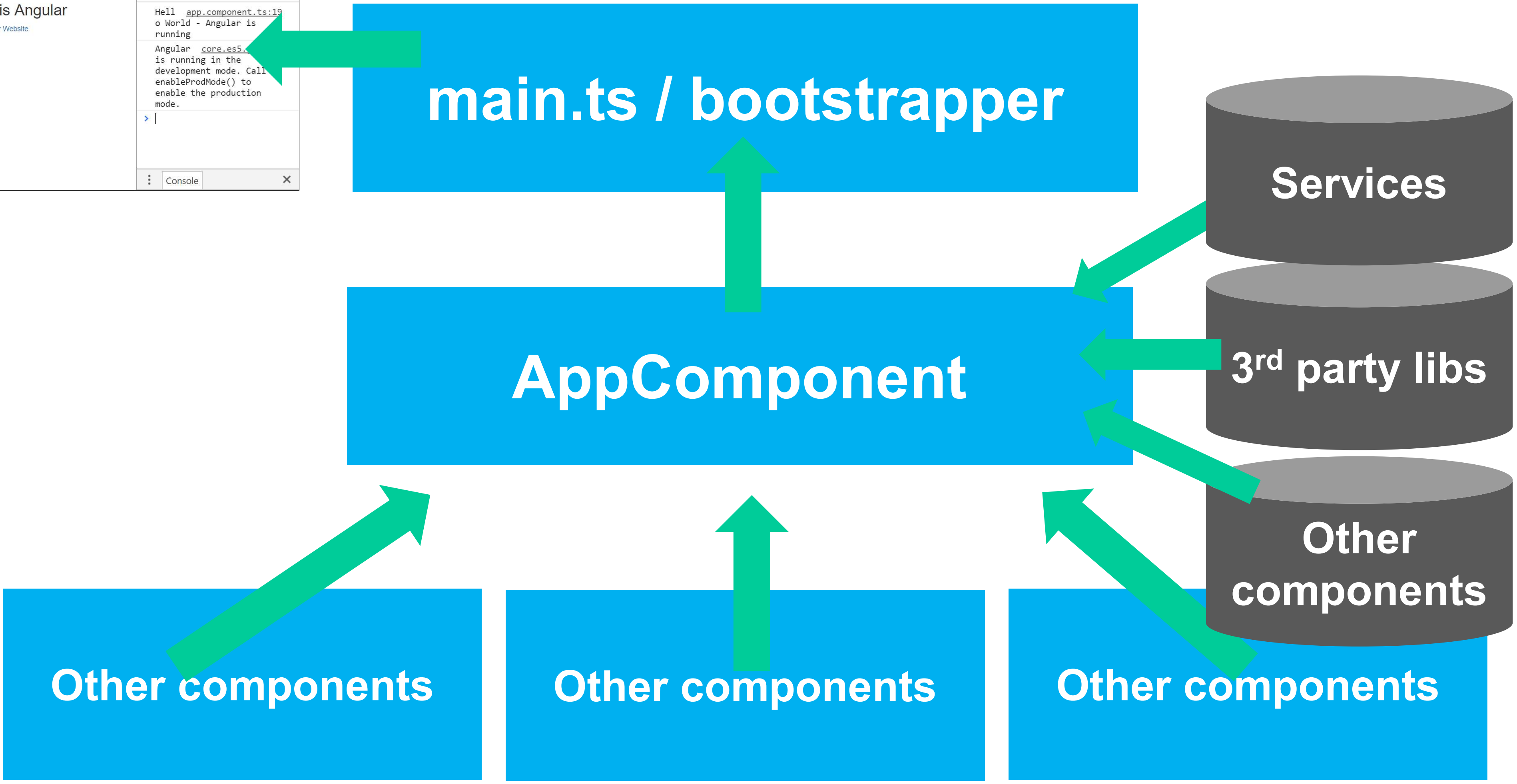


Hello World!

Dit is Angular

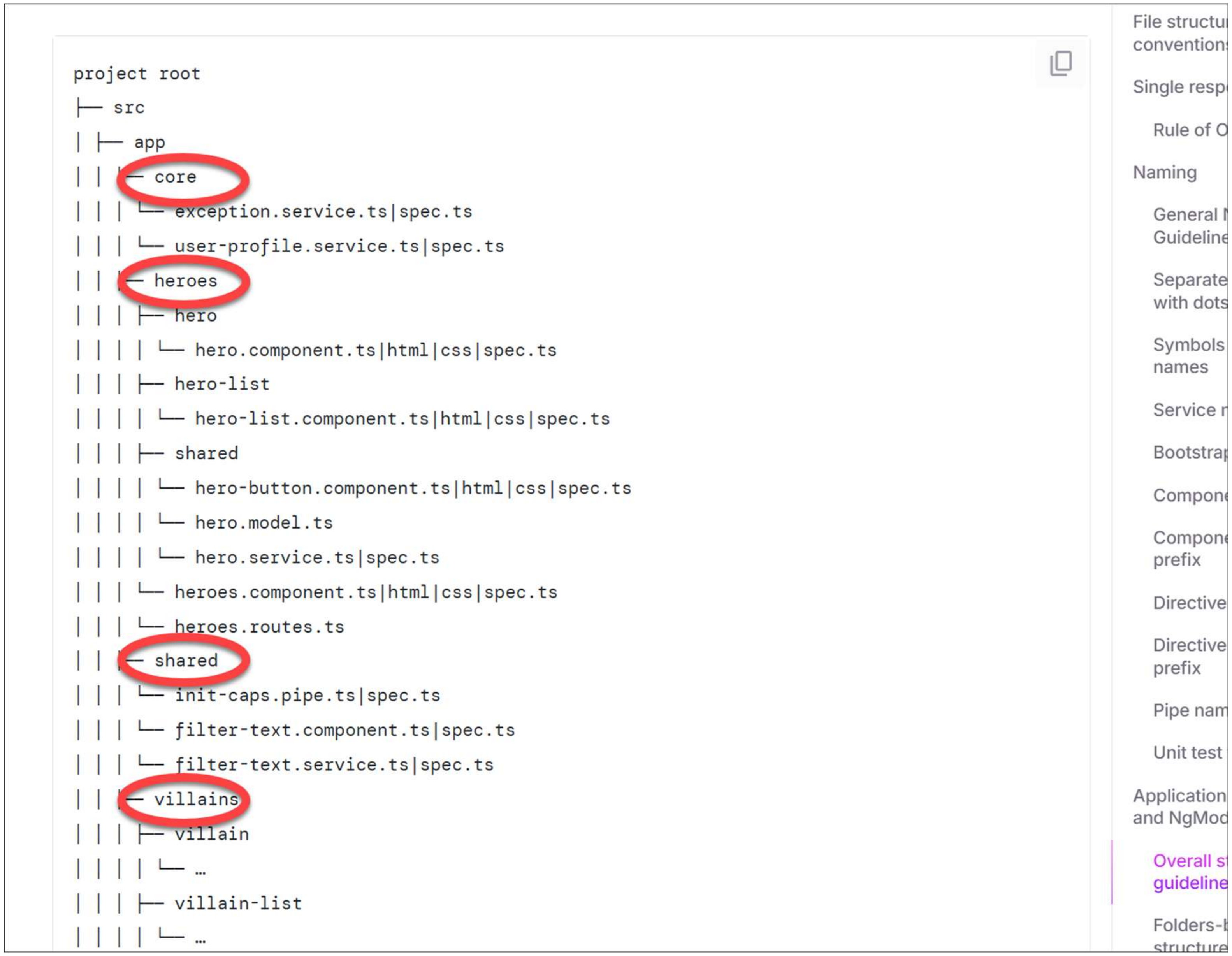
Angular Website

```
Hell  app.component.ts:19
o World - Angular is
running
Angular  core.es5.
is running in the
development mode. Call
enableProdMode() to
enable the production
mode.
```



(Note: no Module anymore, as this is now handled by standalone components)

Angular Coding Style Guide on structure



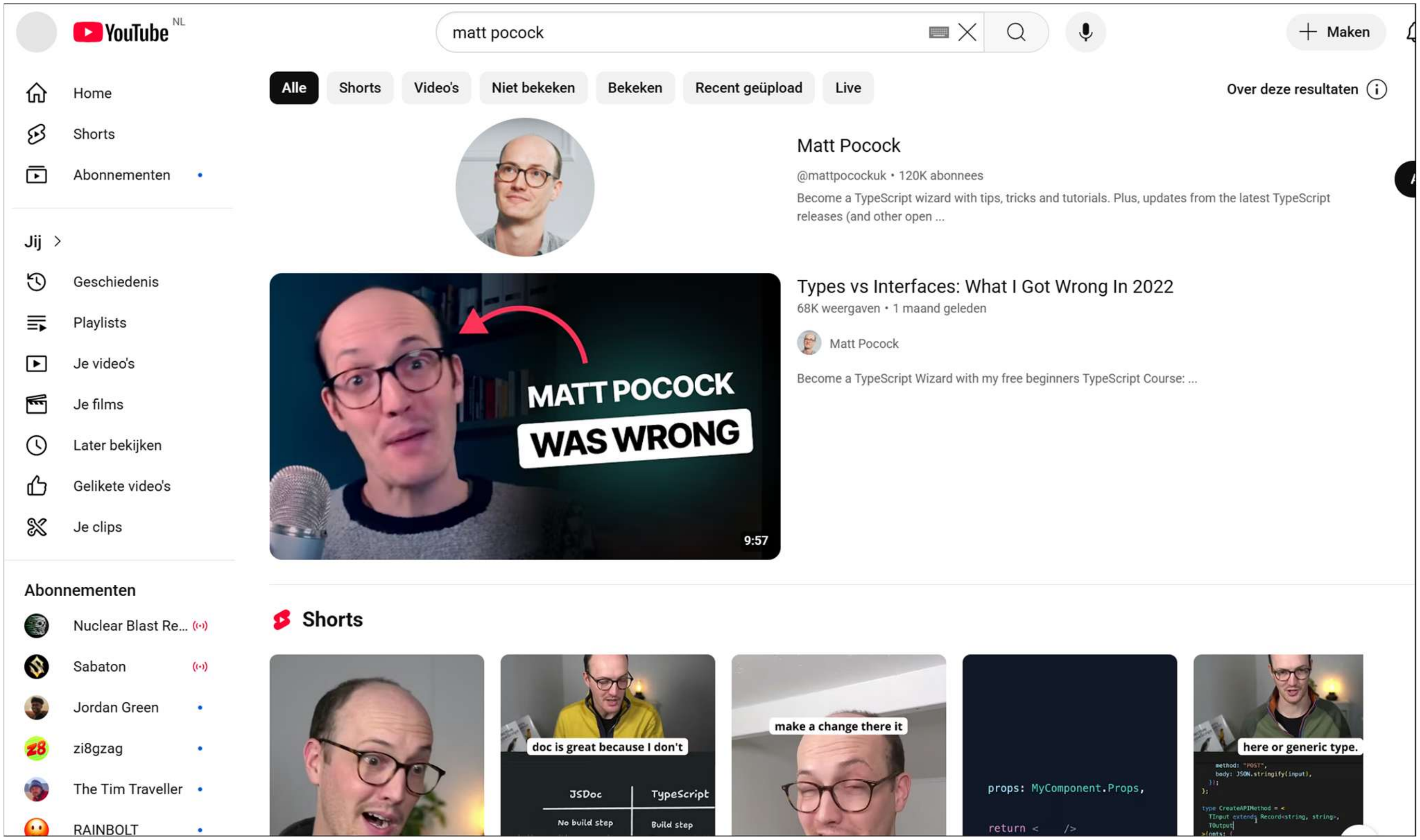
```
project root
├─ src
| ├─ app
| | ├─ core
| | | └─ exception.service.ts|spec.ts
| | | └─ user-profile.service.ts|spec.ts
| | ├─ heroes
| | | ├─ hero
| | | | └─ hero.component.ts|html|css|spec.ts
| | | ├─ hero-list
| | | | └─ hero-list.component.ts|html|css|spec.ts
| | | ├─ shared
| | | | └─ hero-button.component.ts|html|css|spec.ts
| | | | └─ hero.model.ts
| | | | └─ hero.service.ts|spec.ts
| | | └─ heroes.component.ts|html|css|spec.ts
| | | └─ heroes.routes.ts
| | ├─ shared
| | | └─ init-caps.pipe.ts|spec.ts
| | | └─ filter-text.component.ts|spec.ts
| | | └─ filter-text.service.ts|spec.ts
| | ├─ villains
| | | ├─ villain
| | | | └─ …
| | | ├─ villain-list
| | | | └─ …
```

https://angular.dev/style-guide

Useful information

Elsewhere on the internet

TypeScript Guru – Matt Pocock

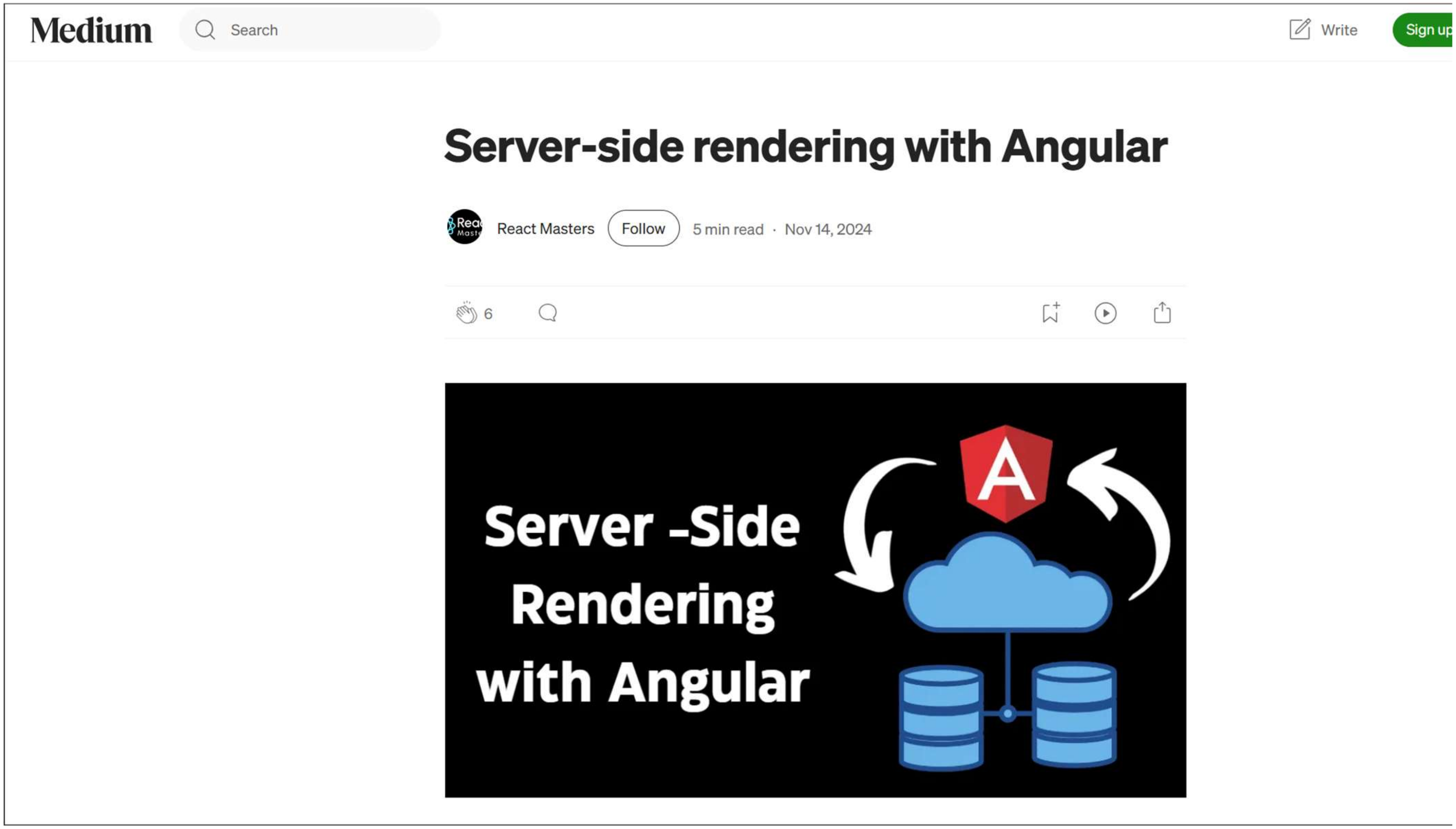


Matt Pocock
@mattpocockuk • 120K abonnees
Become a TypeScript wizard with tips, tricks and tutorials. Plus, updates from the latest TypeScript releases (and other open ...

Types vs Interfaces: What I Got Wrong In 2022
68K weergaven • 1 maand geleden
Matt Pocock
Become a TypeScript Wizard with my free beginners TypeScript Course: ...

https://www.youtube.com/results?search_query=matt+pocock

More info on Angular SSR

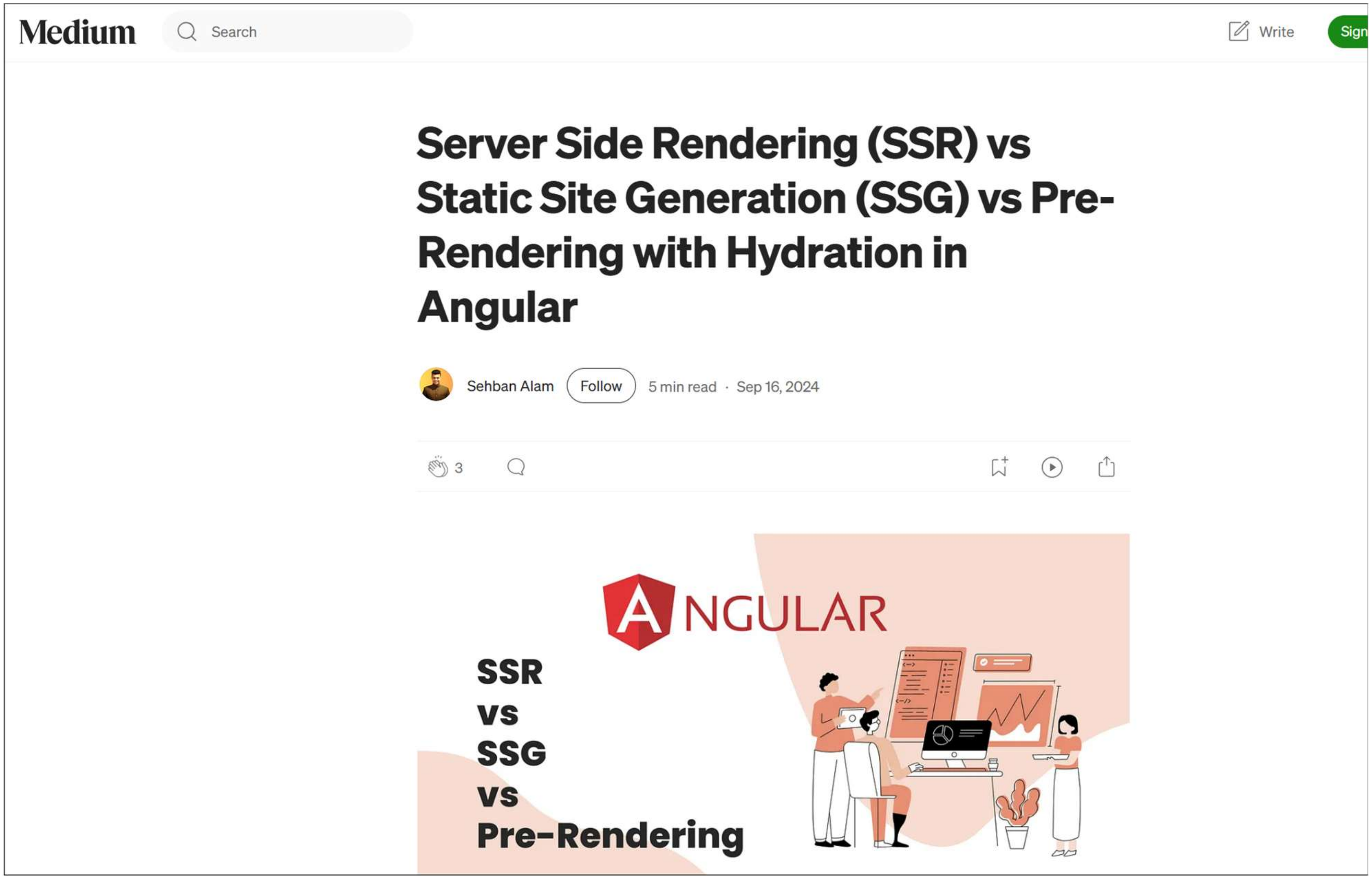


Server-side rendering with Angular

React Masters · Follow · 5 min read · Nov 14, 2024

https://medium.com/@reactmasters.in/server-side-rendering-with-angular-637b0f087c13

More info on SSG



Server Side Rendering (SSR) vs Static Site Generation (SSG) vs Pre-Rendering with Hydration in Angular

Sehban Alam Follow 5 min read · Sep 16, 2024

https://medium.com/@sehban.alam/server-side-rendering-ssr-vs-static-site-generation-ssg-vs-pre-rendering-with-hydration-in-49ff2ef59427

Routing in Angular applications

- Use `<a [routerLink]>` when:
  - You just want a clickable link.
  - Navigation is simple and declarative.
  - No logic needs to be executed before navigation.
  - SEO-friendly, accessible code
- Use `this.router.navigate()` when:
  - Navigation needs to be conditional.
  - You need to run some logic first (e.g., save a form, log out user, etc).
  - You're navigating from a class (e.g., after an API call or in a button handler).
  - Navigating from a service

Simple examples

Using routerLink:

```
<a routerLink="home">Home</a>
```

Navigating from code:

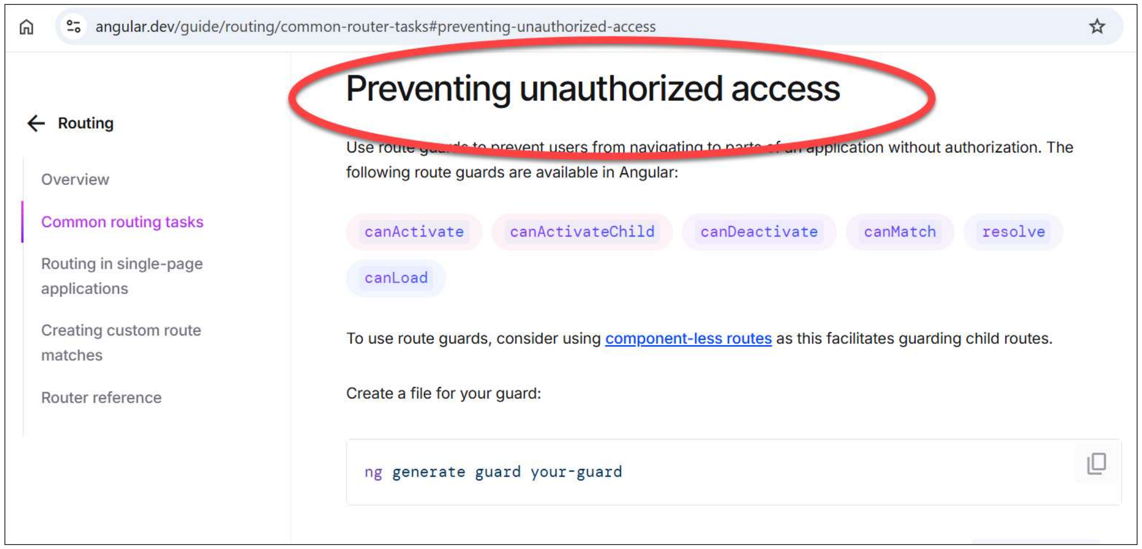
```
constructor(private router: Router) {}

onClick() {
  if (this.form.valid) {
    this.router.navigate(['/home']);
  }
}
```

```
<button (click)="onClick()">Submit and go Home</button>
```

Preventing unauthorized access

- Use Route Guards to prevent users from navigating to parts of an application without authorization.
- The following route guards are available in Angular:
  - `canActivate`
  - `canActivateChild`
  - `canDeactivate`
  - `canMatch`
  - `resolve`
  - `canLoad`


angular.dev/guide/routing/common-router-tasks#preventing-unauthorized-access
Routing
Overview
Common routing tasks
Routing in single-page applications
Creating custom route matches
Router reference

Preventing unauthorized access

Use route guards to prevent users from navigating to parts of an application without authorization. The following route guards are available in Angular:

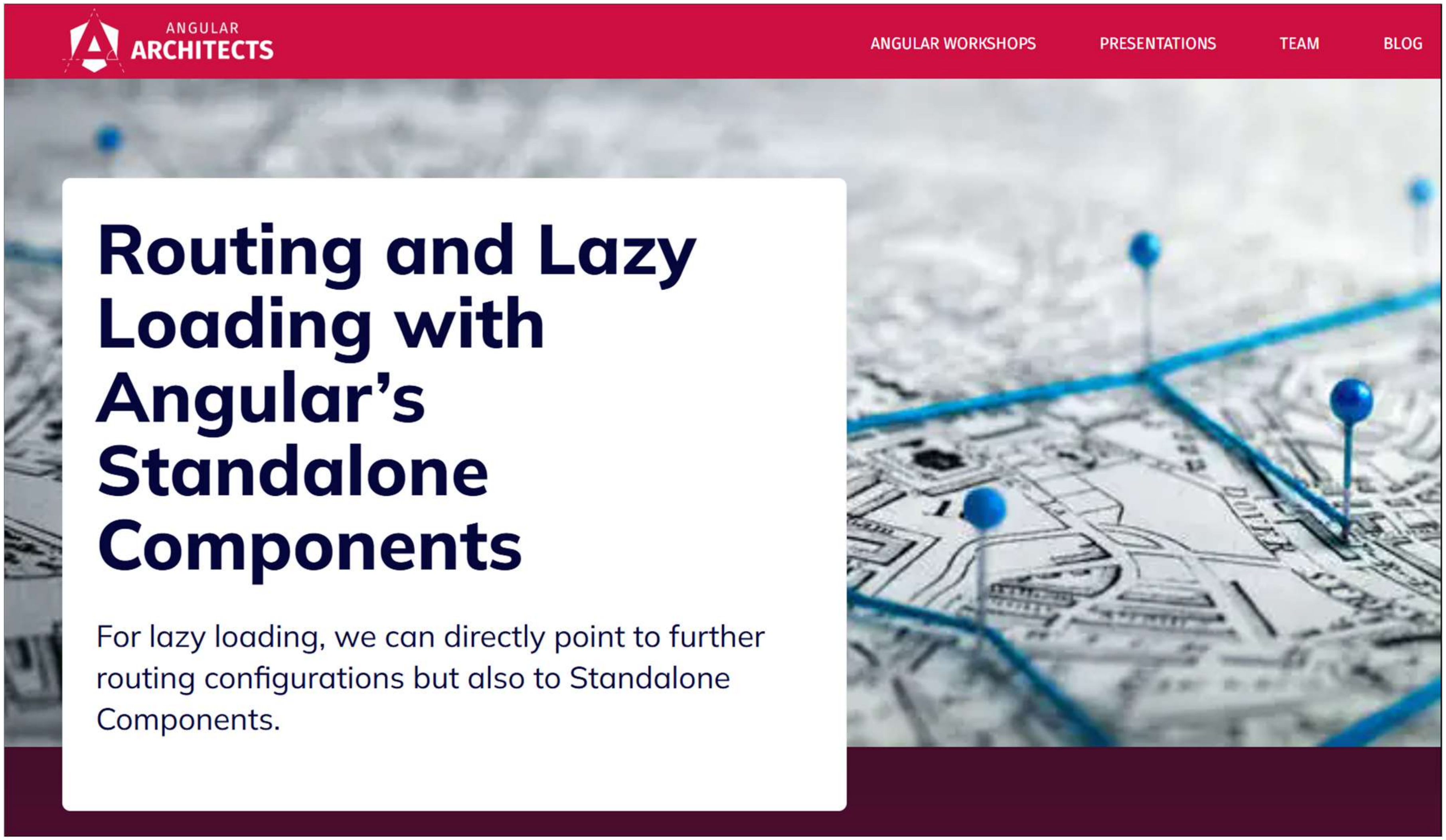
`canActivate` `canActivateChild` `canDeactivate` `canMatch` `resolve` `canLoad`

To use route guards, consider using component-less routes as this facilitates guarding child routes.

Create a file for your guard:

```
ng generate guard your-guard
```

Child Routes



ANGULAR ARCHITECTS

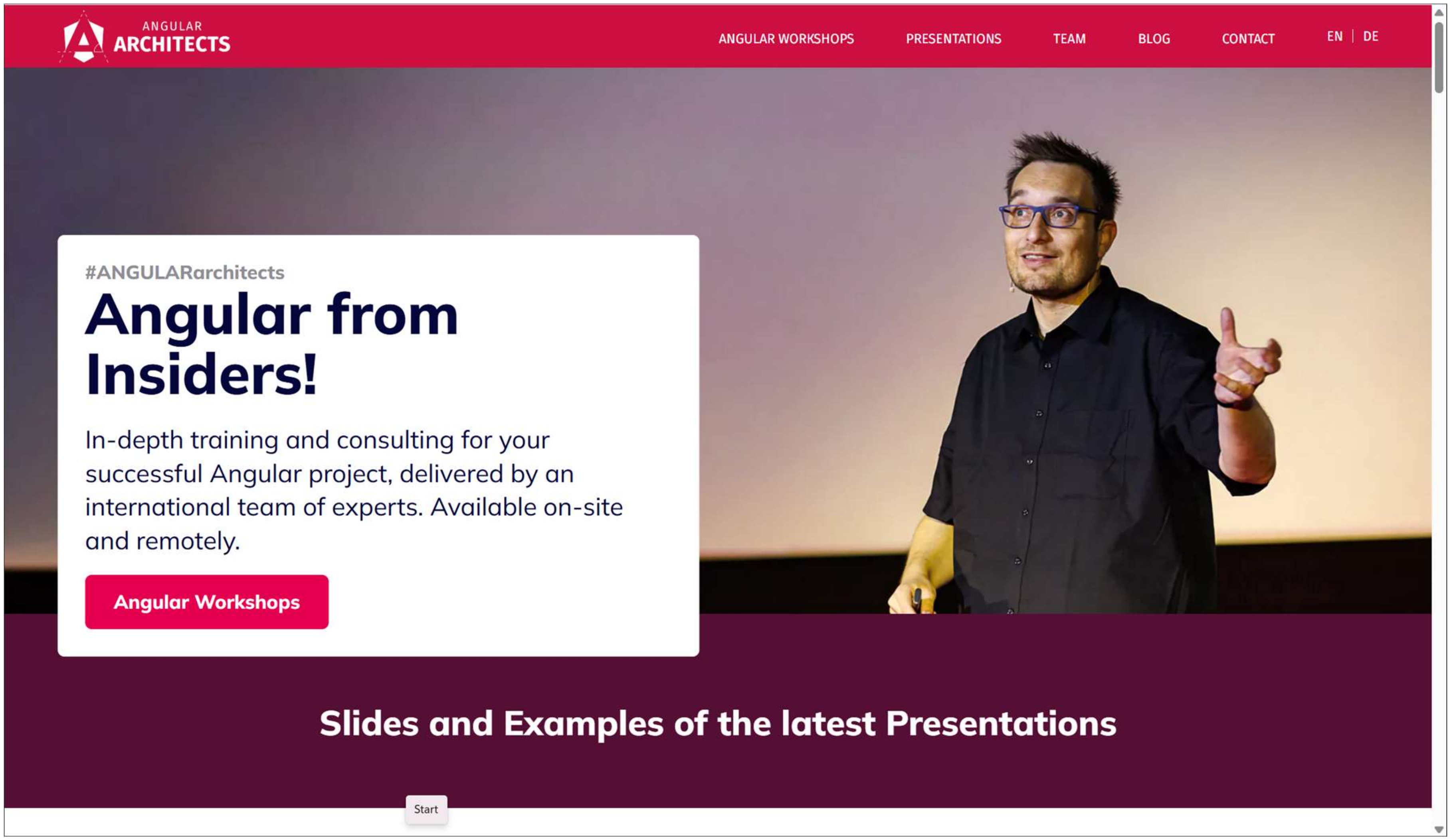
ANGULAR WORKSHOPS PRESENTATIONS TEAM BLOG

Routing and Lazy Loading with Angular's Standalone Components

For lazy loading, we can directly point to further routing configurations but also to Standalone Components.

https://www.angulararchitects.io/en/blog/routing-and-lazy-loading-with-standalone-components/

In general: Manfred Steyer, GDE


ANGULAR ARCHITECTS

ANGULAR WORKSHOPS PRESENTATIONS TEAM BLOG CONTACT EN | DE

#ANGULARarchitects

Angular from Insiders!

In-depth training and consulting for your successful Angular project, delivered by an international team of experts. Available on-site and remotely.

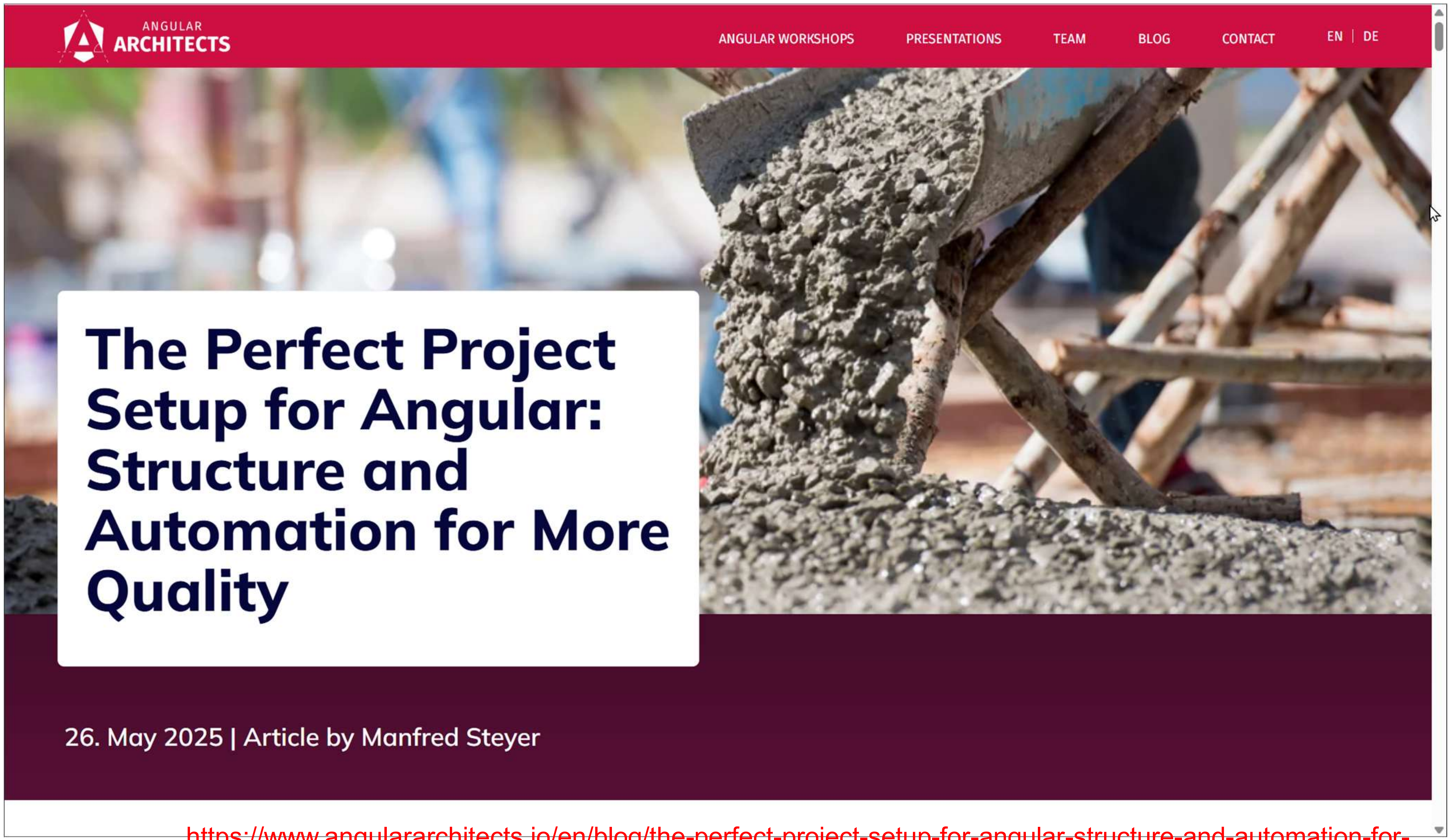
Angular Workshops

Slides and Examples of the latest Presentations

Start

https://www.angulararchitects.io/en/

Article on project structure and file locations



The Perfect Project Setup for Angular: Structure and Automation for More Quality

26. May 2025 | Article by Manfred Steyer

https://www.angulararchitects.io/en/blog/the-perfect-project-setup-for-angular-structure-and-automation-for-more-quality/

Questions?

Agenda - Today, day #2

- State Management
  - General concepts, Redux, NgRX
  - NgRx Effects
- Reactive Form Controls
  - Control states: pristine, dirty, touched, etc.
  - Dynamic Form arrays
- Internationalization
  - Angular Package for i18n
  - Translation files
- …