Global Knowledge®

Angular + TypeScript

Module 1 - Inleiding

Peter Kassenaar –
info@kassenaar.com

Peter Kassenaar

- Trainer, author, developer – since 1996
- Specialty: *"Everything JavaScript"*
- JavaScript, ES6, Angular, NodeJS, TypeScript, React, Vue, Phonegap

www.kassenaar.com

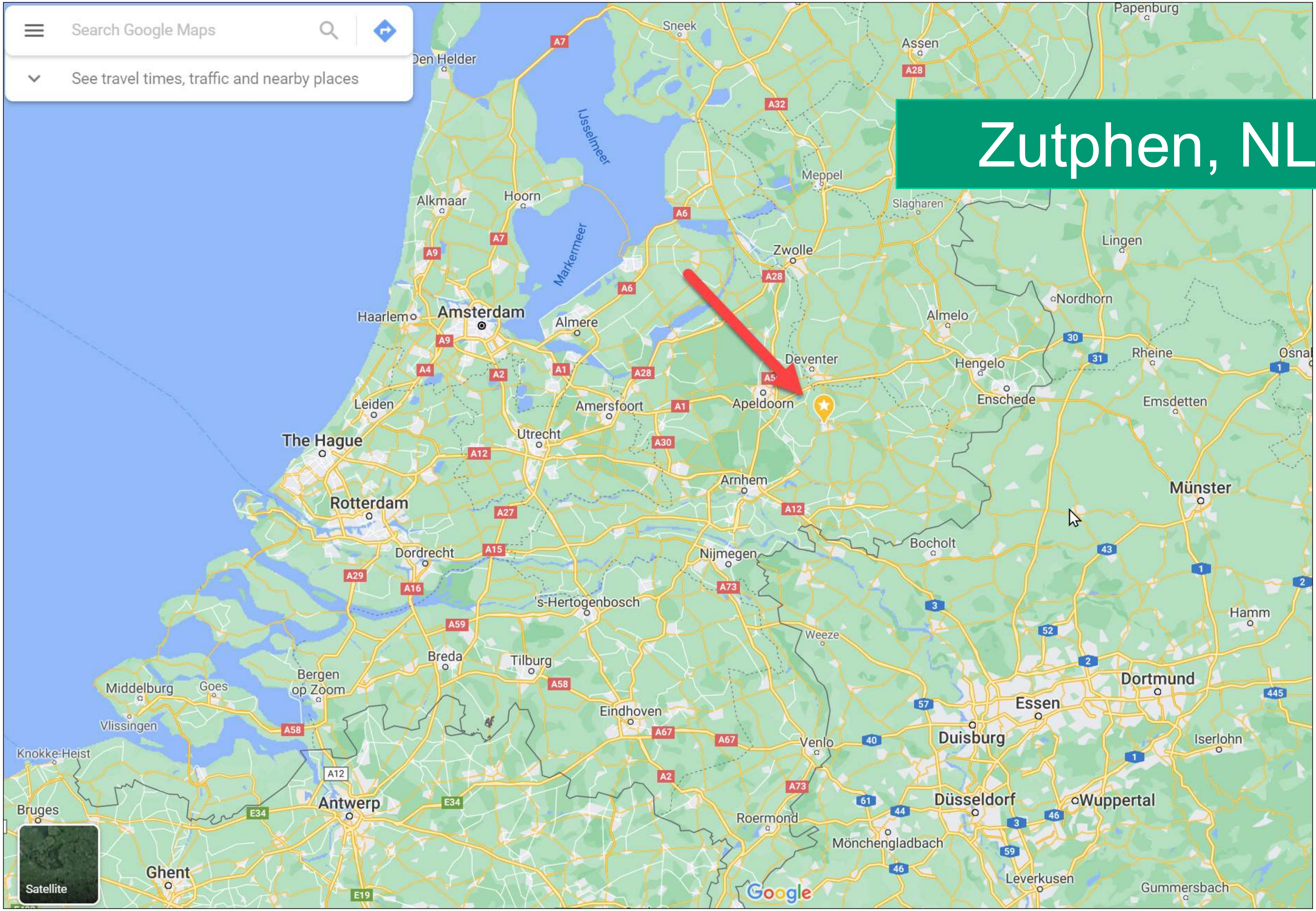
info@kassenaar.com

VANDUUREN MEDIA

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



Search Google Maps
See travel times, traffic and nearby places
Satellite
Google
Den Helder
Sneek
Assen
Papenburg
IJsselmeer
Markermeer
Meppel
Slagharen
Alkmaar
Hoorn
Zwolle
Lingen
Nordhorn
Haarlem
Amsterdam
Almere
Almelo
Deventer
Hengelo
Rheine
Osnal
Enschede
Emsdetten
Leiden
Amersfoort
Apeldoorn
The Hague
Utrecht
Arnhem
Münster
Rotterdam
Bocholt
Dordrecht
Nijmegen
's-Hertogenbosch
Weeze
Hamm
Breda
Tilburg
Bergen op Zoom
Middelburg
Goes
Dortmund
Vlissingen
Eindhoven
Essen
Duisburg
Knokke-Heist
Venlo
Iserlohn
Antwerp
Düsseldorf
Wuppertal
Bruges
Roermond
Mönchengladbach
Ghent
Leverkusen
Gummersbach
A7
A28
A32
A6
A9
A4
A2
A1
A12
A30
A27
A15
A29
A16
A59
A58
A67
A73
E34
E19
30
31
1
43
2
3
52
57
40
445
61
44
46
59

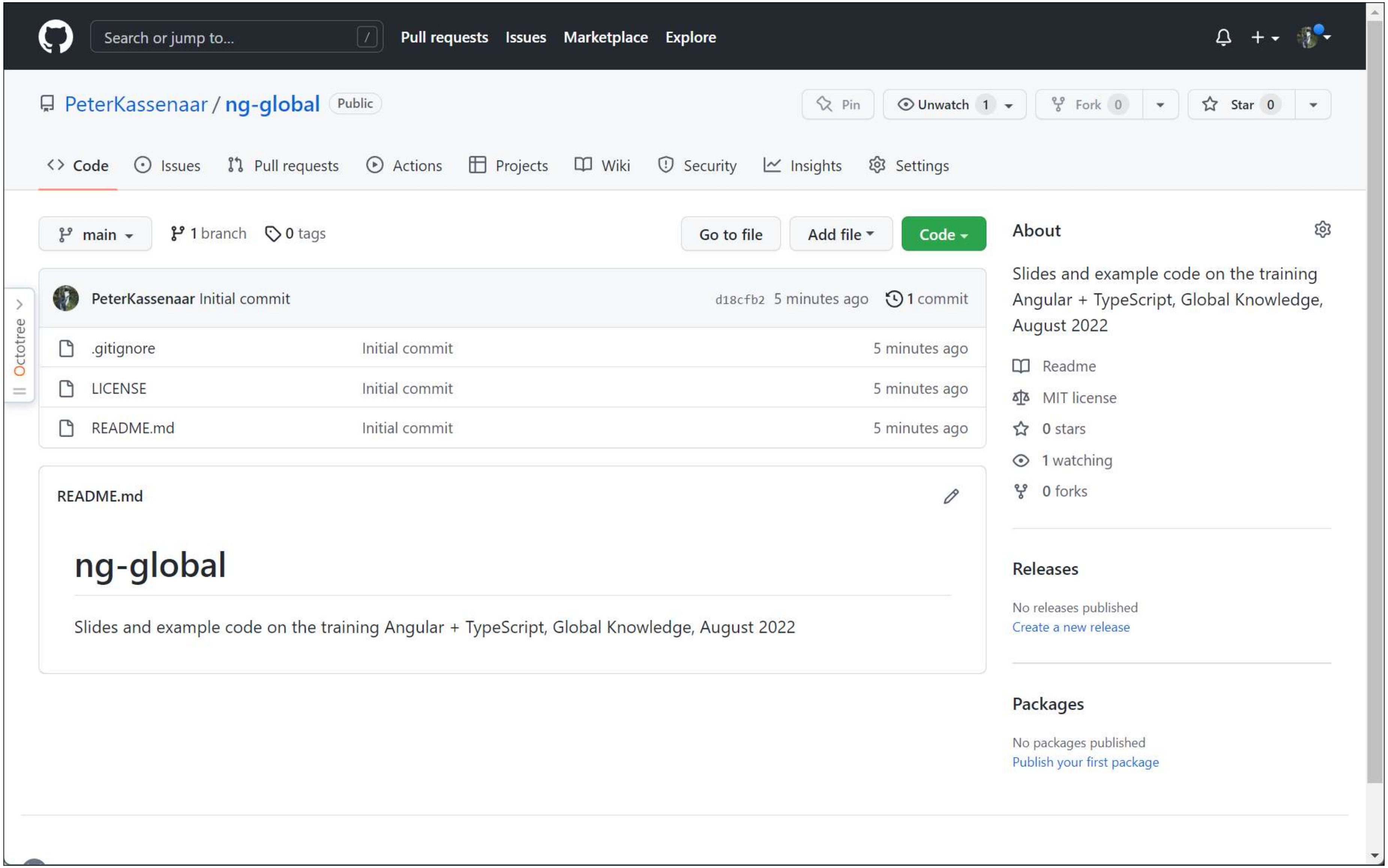
Zutphen, NL

World-class Angular training in Dutch and English

Live classrooms - focused on today's developers

www.angulartraining.nl

github.com/PeterKassenaar/ng-global



PeterKassenaar / ng-global Public

PeterKassenaar Initial commit d18cfb2 5 minutes ago 1 commit

.gitignore	Initial commit	5 minutes ago
LICENSE	Initial commit	5 minutes ago
README.md	Initial commit	5 minutes ago

README.md

ng-global

Slides and example code on the training Angular + TypeScript, Global Knowledge, August 2022

About

Slides and example code on the training Angular + TypeScript, Global Knowledge, August 2022

- 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

Over jullie

Stel jezelf kort voor

Voorkennis webdevelopment, (mobile/web-) apps?

(Kennis AngularJS 1.x?)

Voorkennis andere (web)talen?

Verwachtingen van de cursus?

Concrete projecten?

Specifieke vragen of technieken die je wilt behandelen?

Agenda – globaal – 3 dagen

24 - 26 juli 2022 – wo. – vr.

~ 9:00 start

~ 10:15 short break

~ 12:00 lunch

~ 12:45 middagsessie

~ 14:15 short break

~16:00 Einde

Doel van de training

*Je wordt **geen** Angular wizard in 3 dagen (sorry)*

maar….

Doelen

1. Je leert over de structuur en architectuur van Angular Apps. Van een kleine `hello-world` app tot een grote Enterprise applicatie.

2. Je bent bekend met de belangrijkste **Angular concepten** van het framework. Specifieke details kun je altijd Googelen.

3. Je hebt enige hands-on ervaring met het maken van apps en componenten, services, API's/backends, component communicatie.

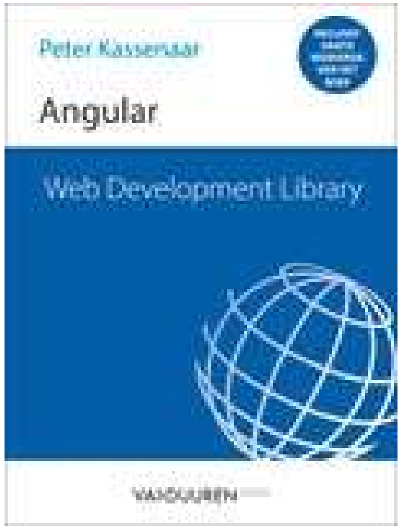
4. Je hebt een algemeen begrip van de manier waarop moderne web apps worden gemaakt met Angular, TypeScript en build tools.

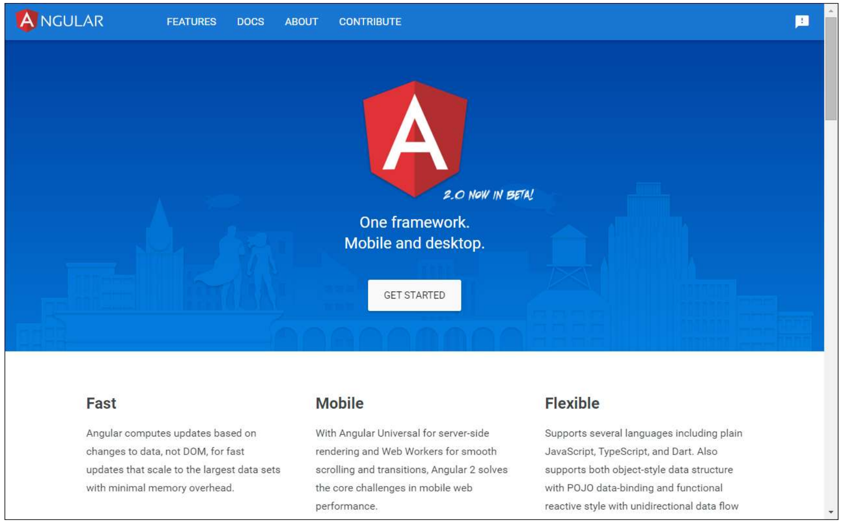
Agenda - 3 dagen

- Introductie & geschiedenis - waarom Angular, TypeScript?
- Kernbegrippen in Angular 2 – 13
- CLI, Hello World in Angular – inzicht in boilerplate-code
- Angular in depth (modules):
  - Components
  - ECMAScript 2015 + TypeScript
  - Data binding
  - Dependency Injection (DI) – more components
  - Services en http, Observables (RxJS), communicatie backend
  - Intercomponent communication
- BEST PRACTICES / STYLE GUIDE

Materialen

Software	(Angular, NodeJS + NPM, editor, browser)
Handouts	(Github - PDF)
Oefeningen	(Github)
Websites	(online)





One framework.
Mobile and desktop.

Fast

Angular computes updates based on changes to data, not DOM, for fast updates that scale to the largest data sets with minimal memory overhead.

Mobile

With Angular Universal for server-side rendering and Web Workers for smooth scrolling and transitions, Angular 2 solves the core challenges in mobile web performance.

Flexible

Supports several languages including plain JavaScript, TypeScript, and Dart. Also supports both object-style data structure with POJO data-binding and functional reactive style with unidirectional data flow

angular.io/

Vandaag

- Dag 1 – Intro & Data binding
  - Theorie - Introductie & geschiedenis – waarom Angular
  - Hello World in Angular –boilerplate-code
    - Components + Modules
  - Concepts, context & architecture
  - Angular CLI, TypeScript
  - Data binding

2 Richtlijnen

1. Oefeningen
   - Maar: neem ook vooral zijpaden, experimenteer, lees verder, maak een eigen project, app, website…

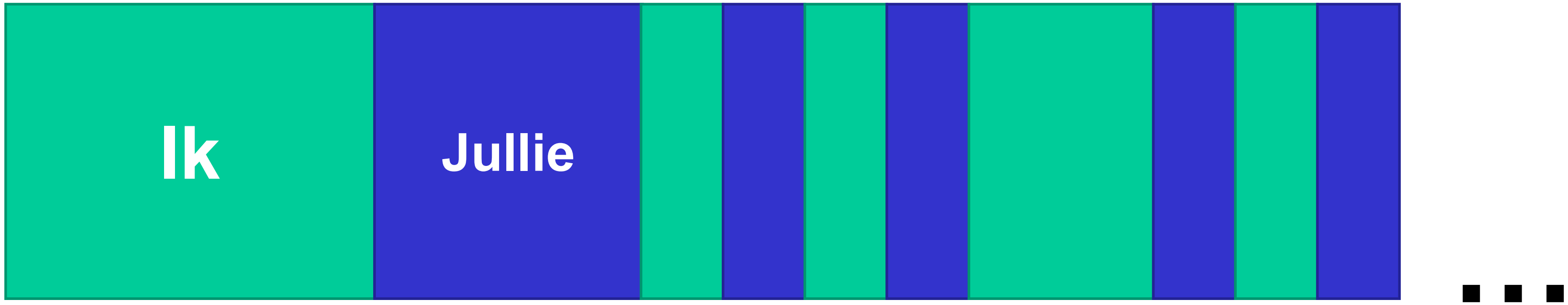
2. Voorbeeldcode
   - Als ondersteuning bij de oefeningen, zie boven
   - Work in progress – check de Angular-site!
   - github.com/PeterKassenaar/voorbeeldenAngular2

Advanced warning - First morning

Not so much code…

Concepts, architecture, structure

Globale werkwijze



Vragen?

Angular vs. The Rest

Differences, similarities, new features

Addressing the "WHY" question!

WHY, would we want to use a frontend framework.

It is all HTML, CSS and JavaScript right?

Rethorical question:

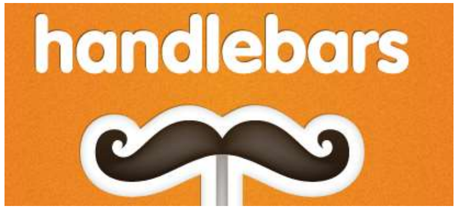
"Do we want to go back

to the jQuery days?"

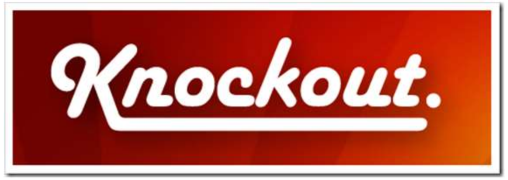
speed, consistency, not re-inventing the wheel, community, performance, testing….

Old school web apps

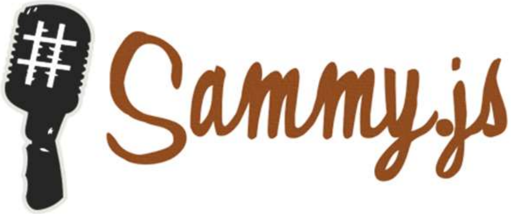
HTML + templates



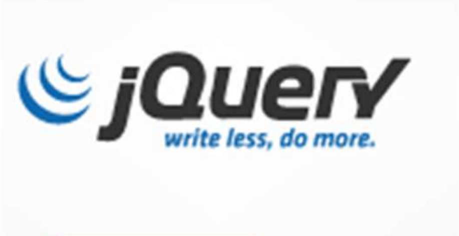
Data Binding



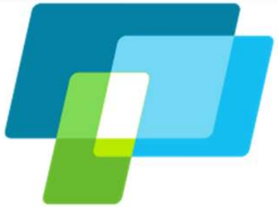
Routing



DOM-manipulation



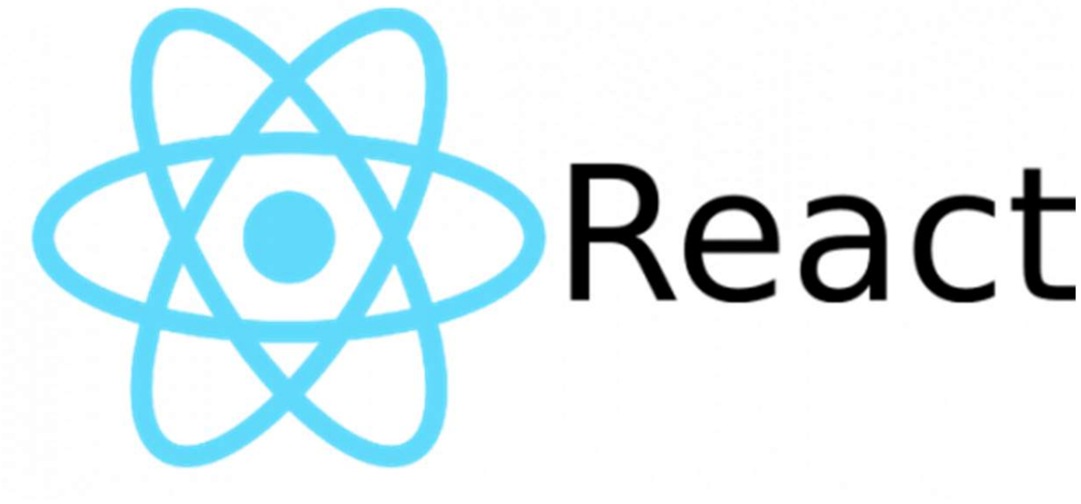
Mobile development



...

“The Frankenstein Framework”

Front-end Frameworks – the big four



Similarities

- Creating Single Page Applications
- Based on components
- Data binding, props, events, routing, state management, …
- Huge ecosystem
- Huge community
- High adaptation rate

Differences (apart from syntax)

- Point of departure: HTML template, enhanced with framework specific tags and attributes
- One-stop-shop / solution

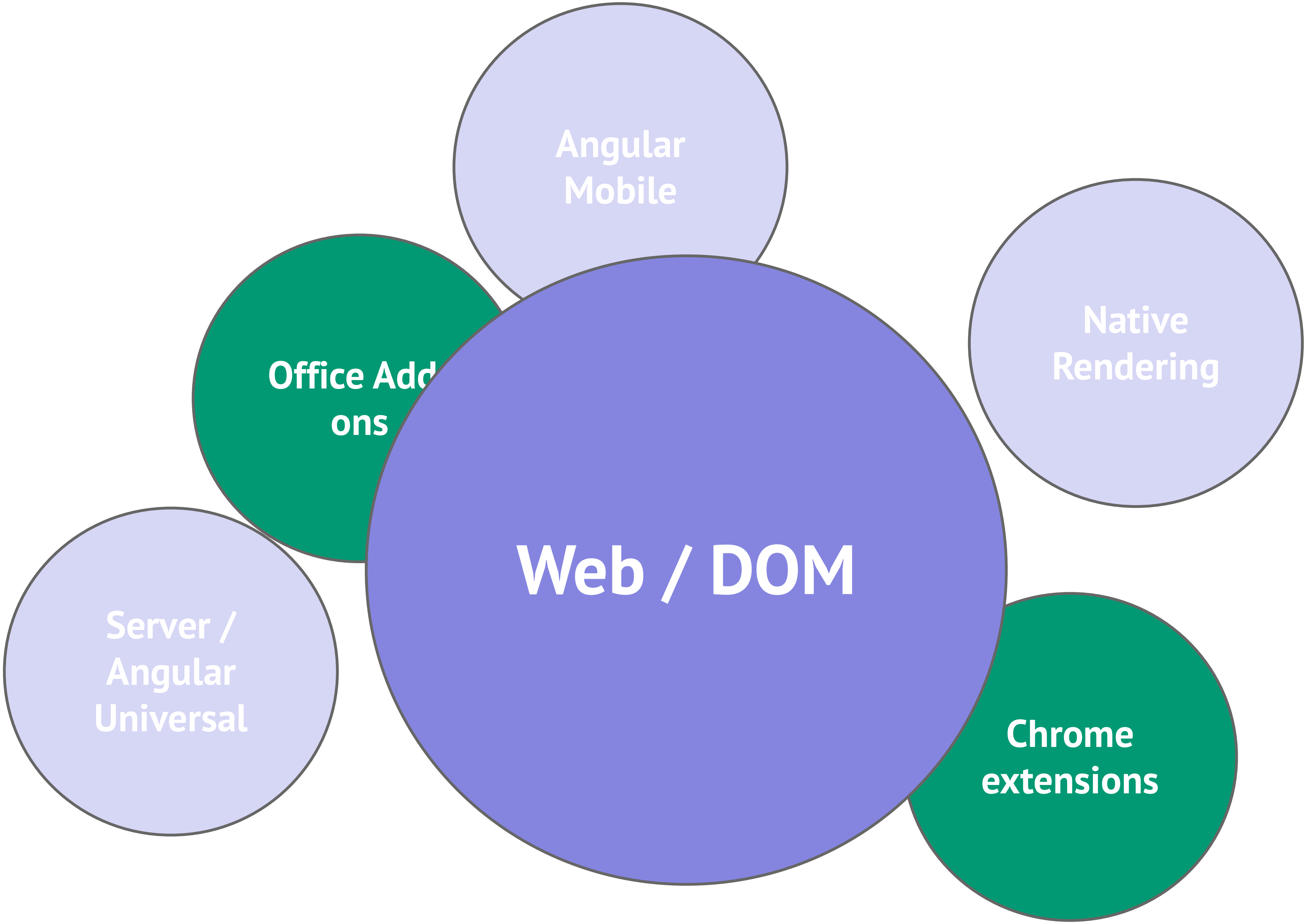
React

- Point of departure: JavaScript, JavaScript, JavaScript (JSX)
- Build-all-yourself / choice anxiety

Platform

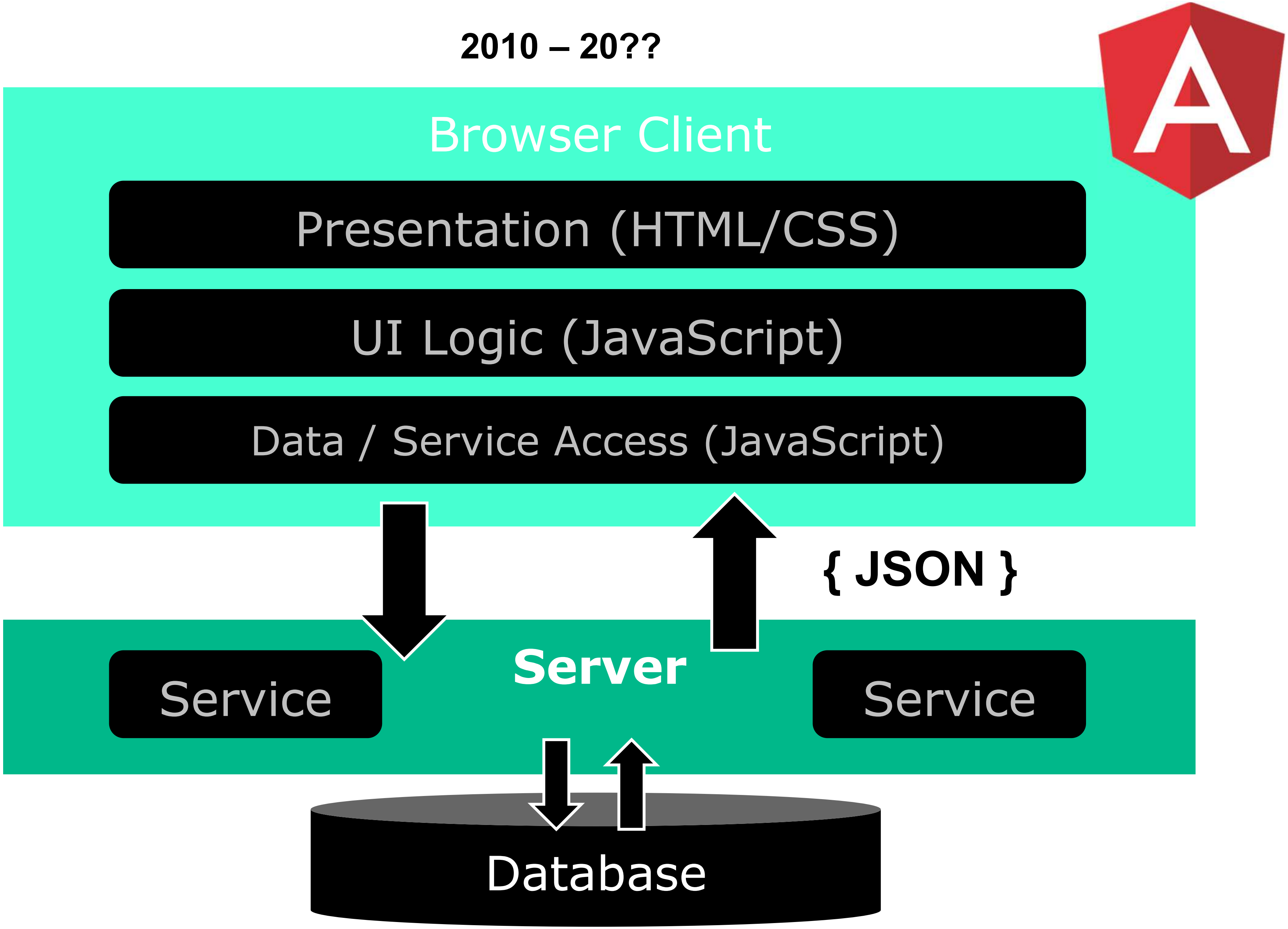
Platform Features

	Scaffolding	Code completion & Refactoring	Debugging
Tooling	Angular CLI	Language Services	DevTools
Libraries	Material 2	Mobile	Universal
	AOT- Compile	Change Detection	Renderer
Core	Components & Dependency Injection	Decorators	Zones


Angular Mobile
Office Add ons
Native Rendering
Web / DOM
Server / Angular Universal
Chrome extensions

Single Page Application

2010 – 20??



Browser Client

Presentation (HTML/CSS)

UI Logic (JavaScript)

Data / Service Access (JavaScript)

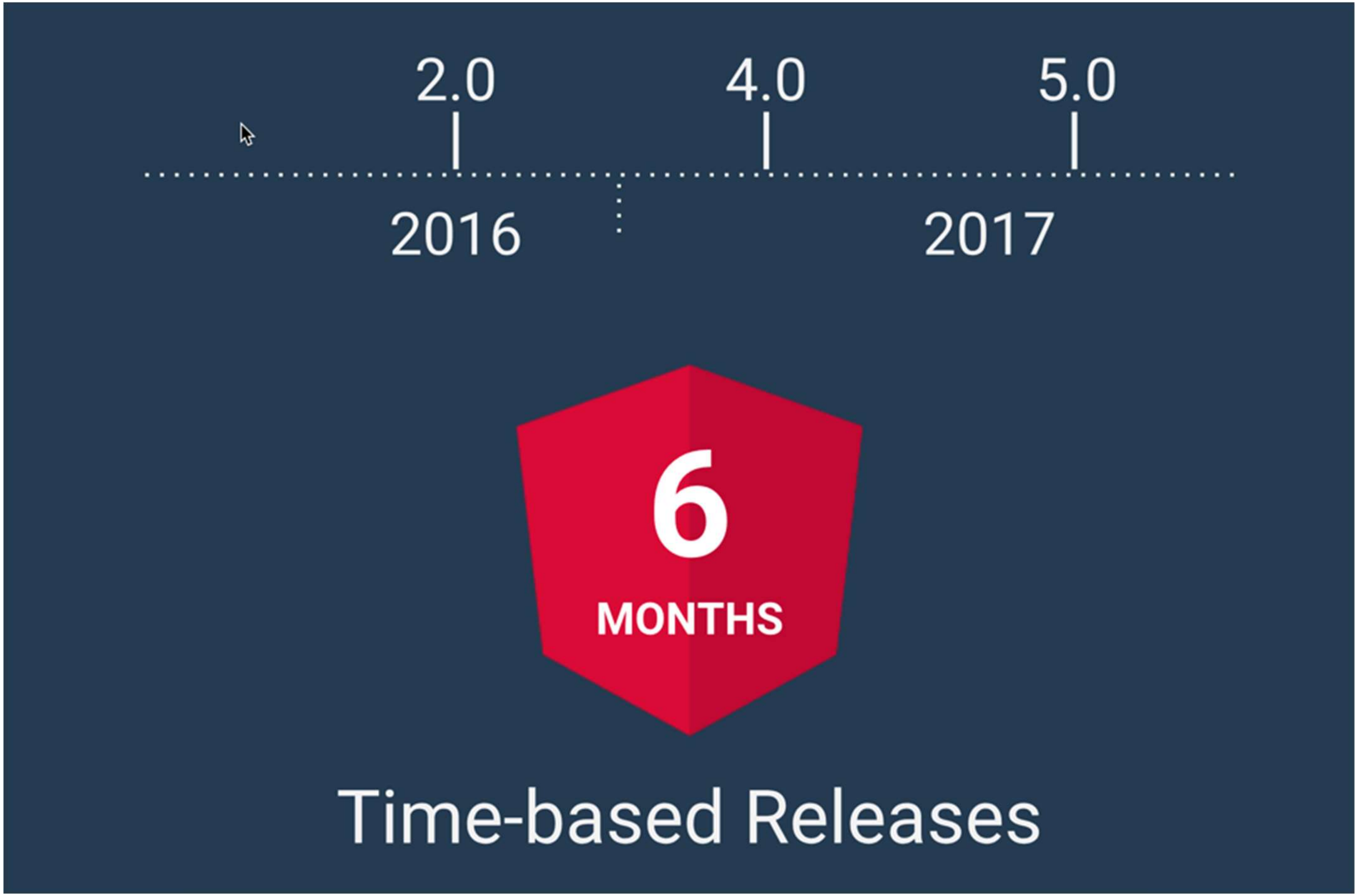
{ JSON }

Server

Service

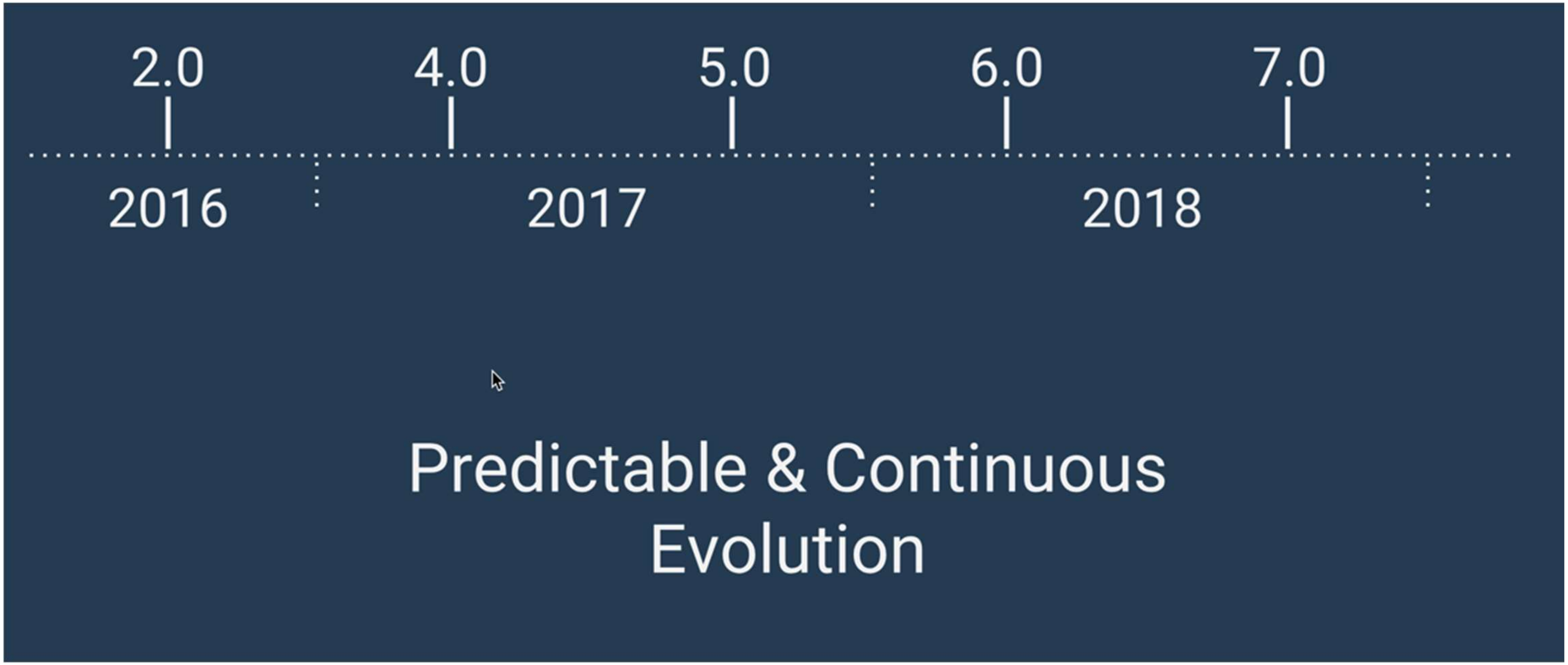
Service

Database



Time-based Releases

@IgorMinar, Nov. 9, 2018



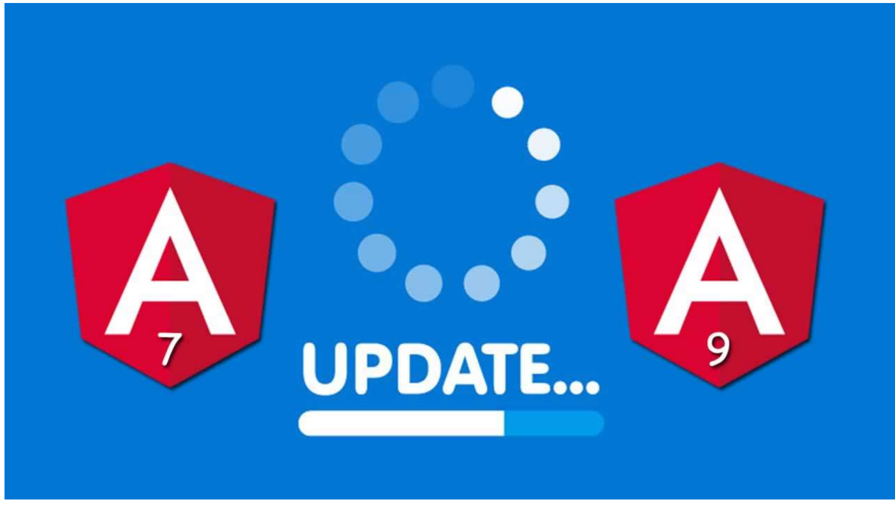
2.0 4.0 5.0 6.0 7.0

2016 2017 2018

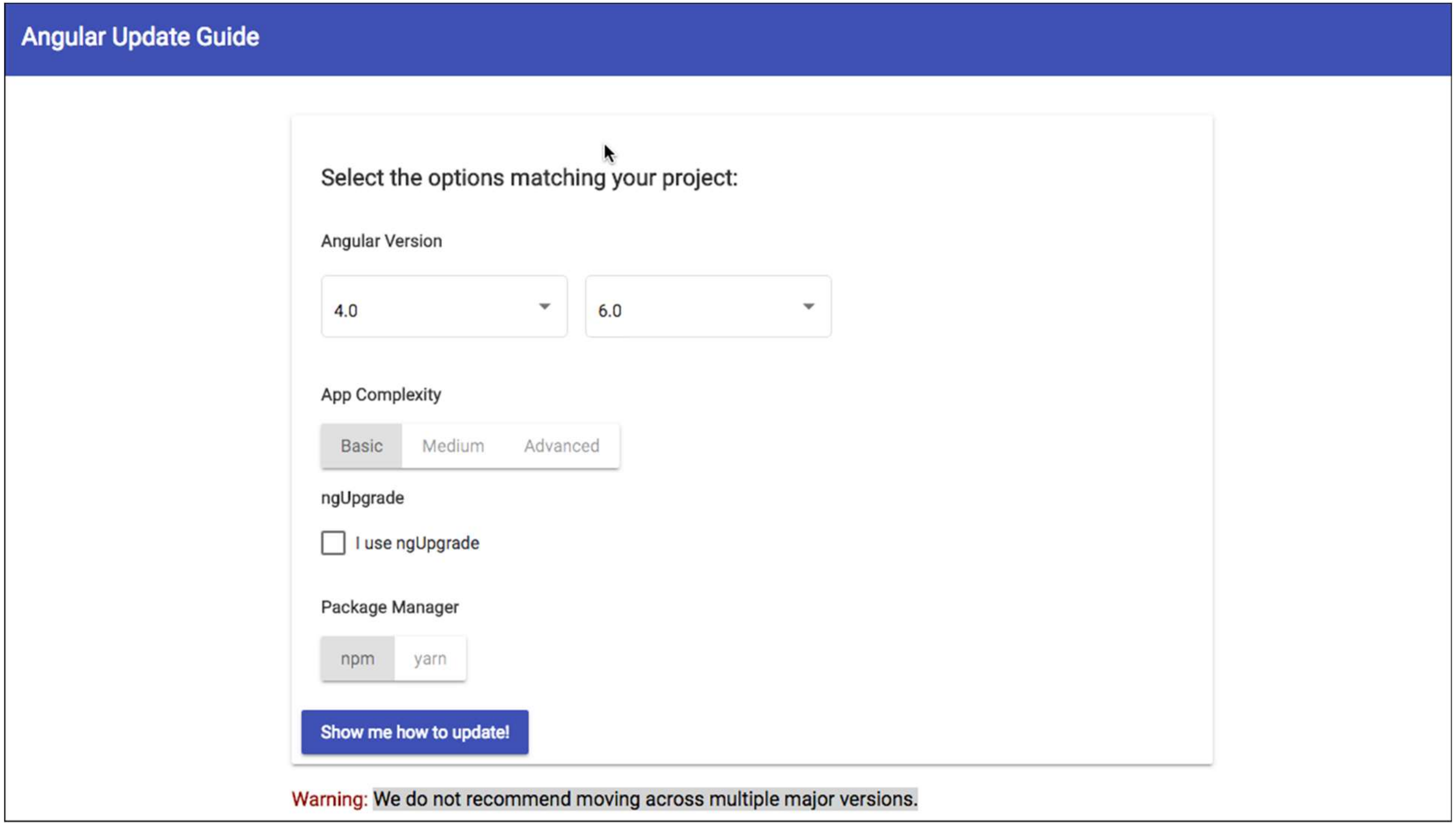
Predictable & Continuous Evolution

Angular updates

- Angular 2 – 14 **Grotendeels 'meer van hetzelfde'**
  - Soms zijn er breaking changes
  - Zelf checken: https://update.angular.io/
  - Updates van verwante libraries (RxJS, Angular Material)
- Belangrijke milestones
  - Angular 2 – breekt met structuur en architectuur van Angular JS (1.x)
  - Angular 6 – Introductie modern Angular CLI
  - Angular 9 – Ivy Compiler
  - Angular 14 – Standalone components



https://update.angular.io/



Angular Update Guide

Select the options matching your project:

Angular Version

4.0 ▾ 6.0 ▾

App Complexity

[✓ Basic] Medium Advanced

ngUpgrade

□ I use ngUpgrade

Package Manager

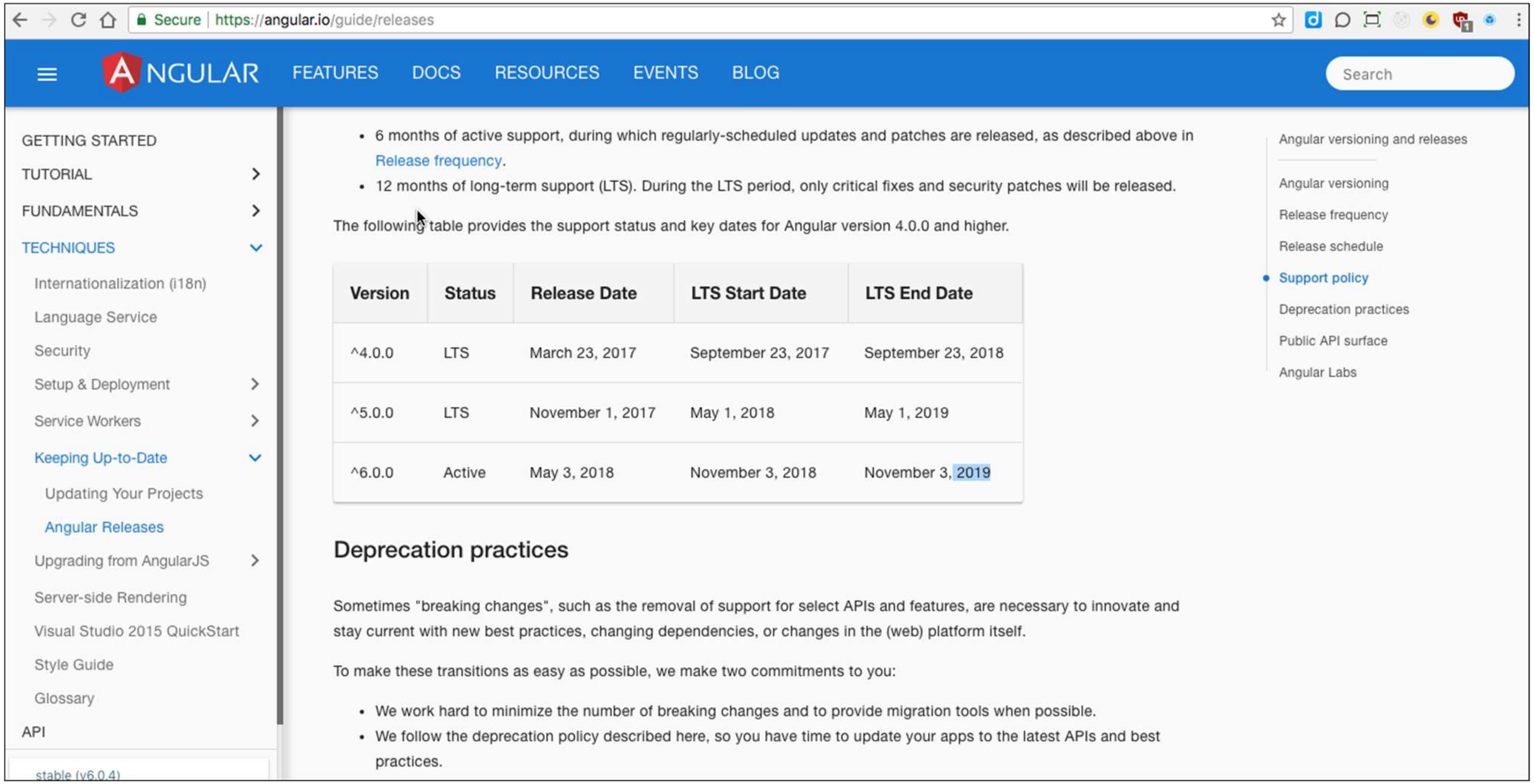
[✓ npm] yarn

Show me how to update!

Warning: We do not recommend moving across multiple major versions.

Angular Versies en -Long Time Support

→ https://angular.io/guide/releases



- 6 months of active support, during which regularly-scheduled updates and patches are released, as described above in Release frequency.
- 12 months of long-term support (LTS). During the LTS period, only critical fixes and security patches will be released.

The following table provides the support status and key dates for Angular version 4.0.0 and higher.

Version	Status	Release Date	LTS Start Date	LTS End Date
^4.0.0	LTS	March 23, 2017	September 23, 2017	September 23, 2018
^5.0.0	LTS	November 1, 2017	May 1, 2018	May 1, 2019
^6.0.0	Active	May 3, 2018	November 3, 2018	November 3, 2019

Deprecation practices

Sometimes "breaking changes", such as the removal of support for select APIs and features, are necessary to innovate and stay current with new best practices, changing dependencies, or changes in the (web) platform itself.

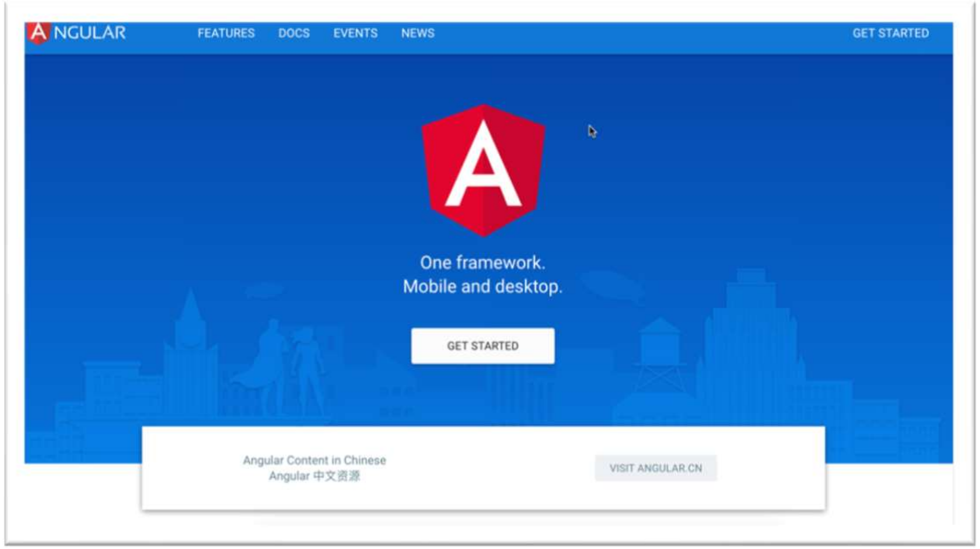
To make these transitions as easy as possible, we make two commitments to you:

- We work hard to minimize the number of breaking changes and to provide migration tools when possible.
- We follow the deprecation policy described here, so you have time to update your apps to the latest APIs and best practices.

“It's just

Angular”

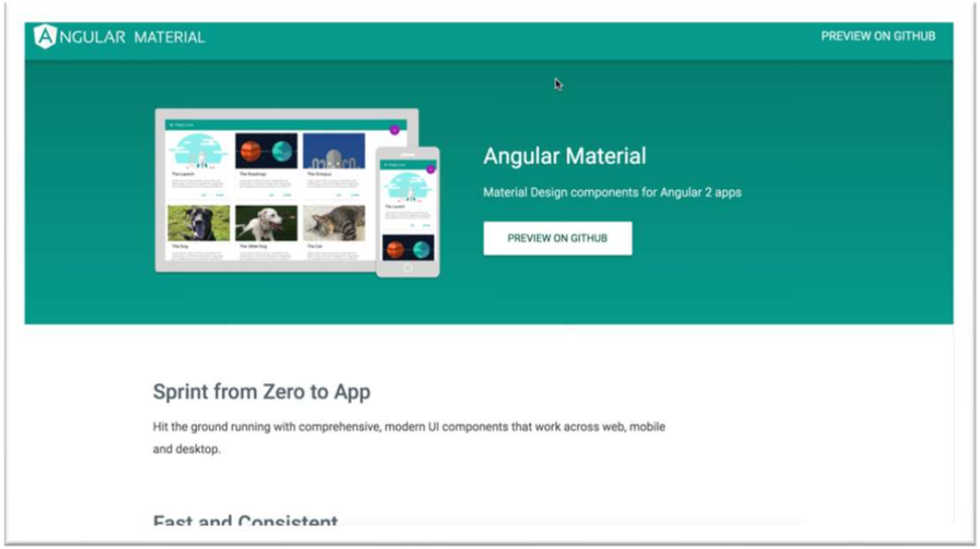
Angular as a Platform



One framework.
Mobile and desktop.

Angular Content in Chinese
Angular 中文资源

https://angular.io/



Angular Material

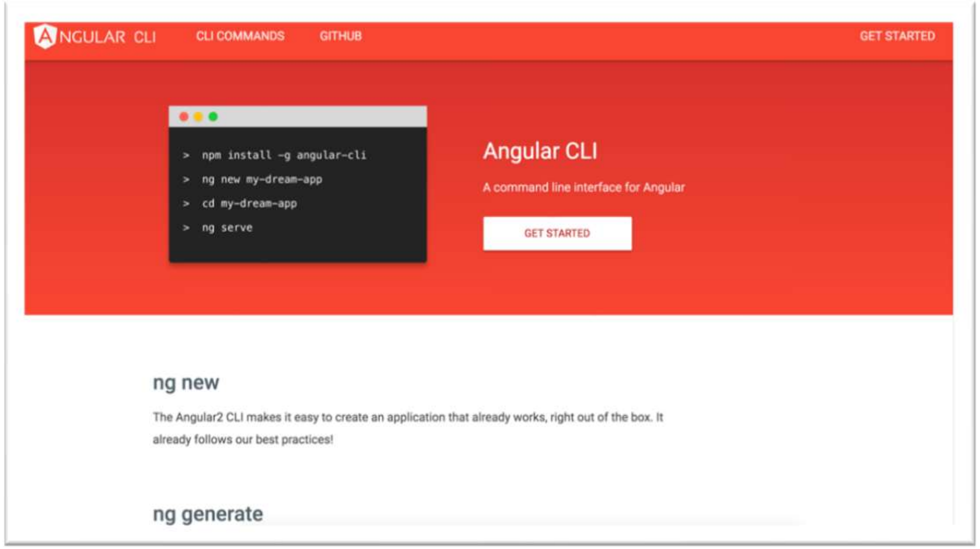
Material Design components for Angular 2 apps

Sprint from Zero to App

Hit the ground running with comprehensive, modern UI components that work across web, mobile and desktop.

Fast and Consistent

https://material.angular.io/



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

Angular CLI

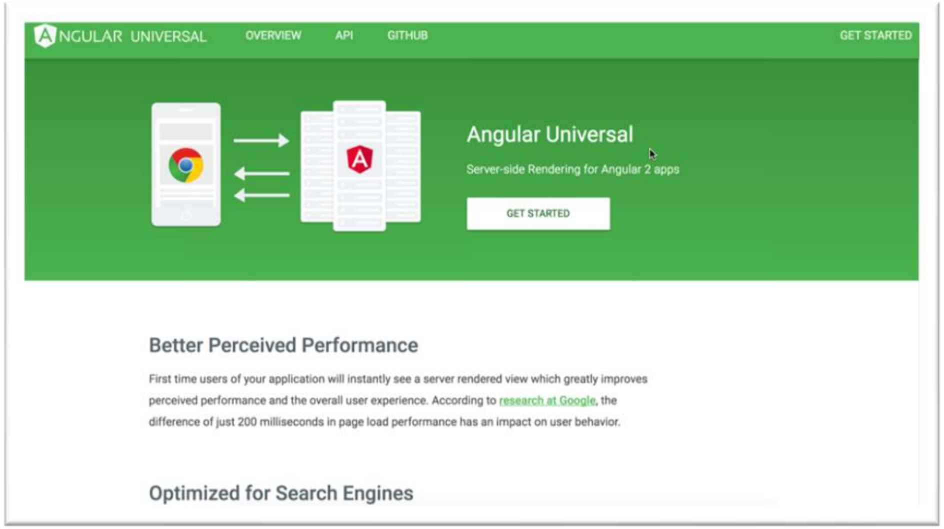
A command line interface for Angular

ng new

The Angular2 CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

ng generate

https://cli.angular.io/



Angular Universal

Server-side Rendering for Angular 2 apps

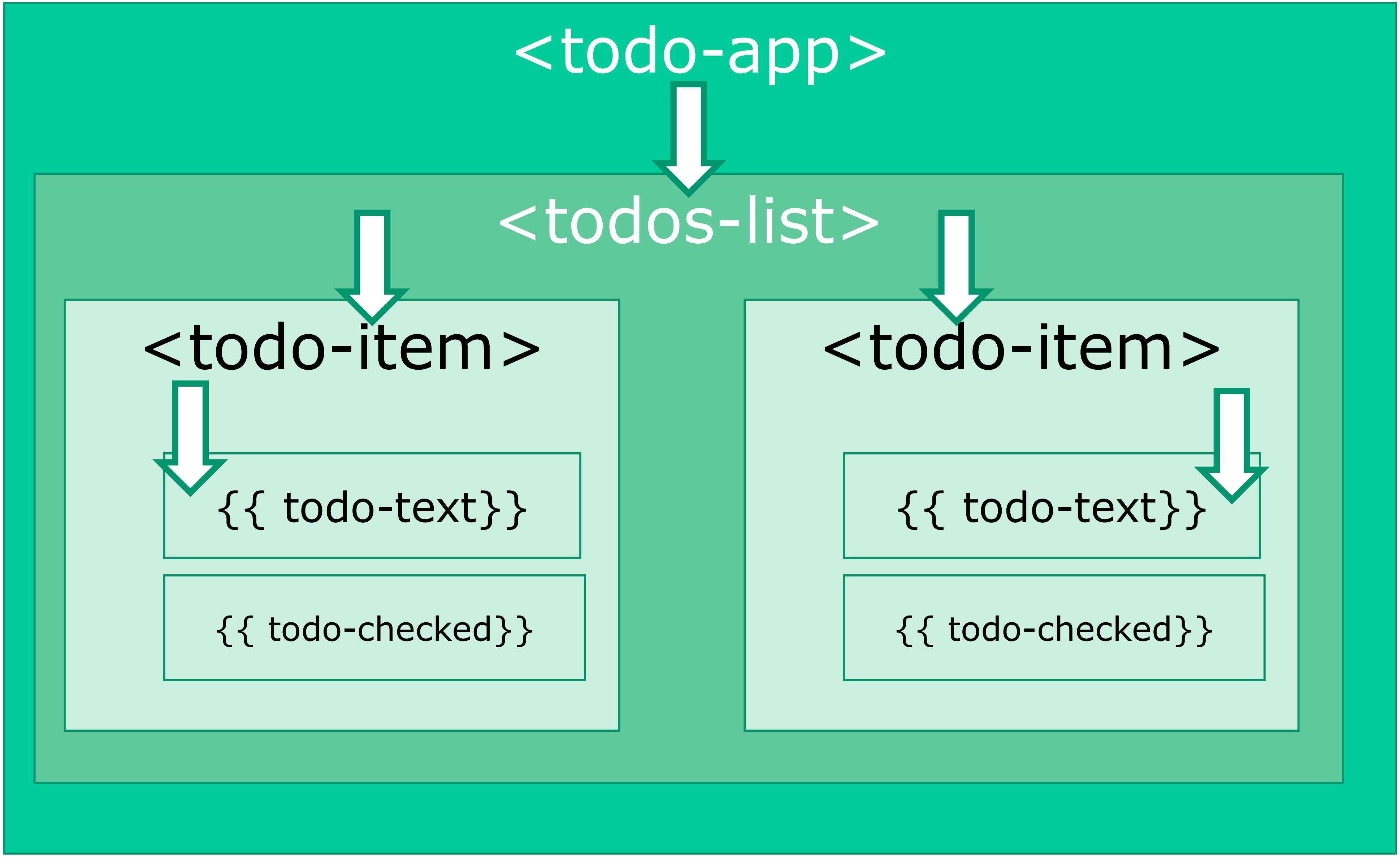
Better Perceived Performance

First time users of your application will instantly see a server rendered view which greatly improves perceived performance and the overall user experience. According to research at Google, the difference of just 200 milliseconds in page load performance has an impact on user behavior.

Optimized for Search Engines

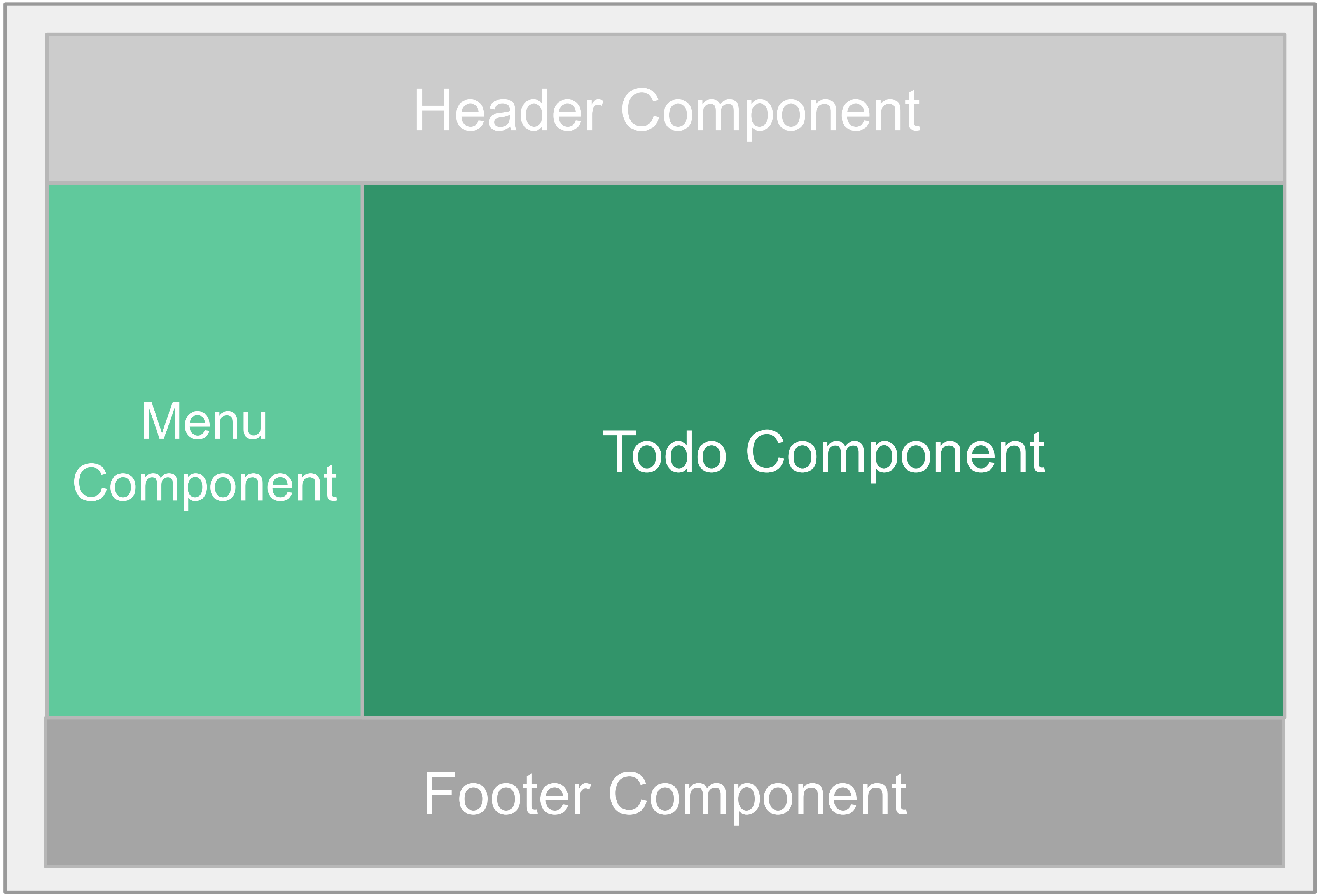
https://universal.angular.io/

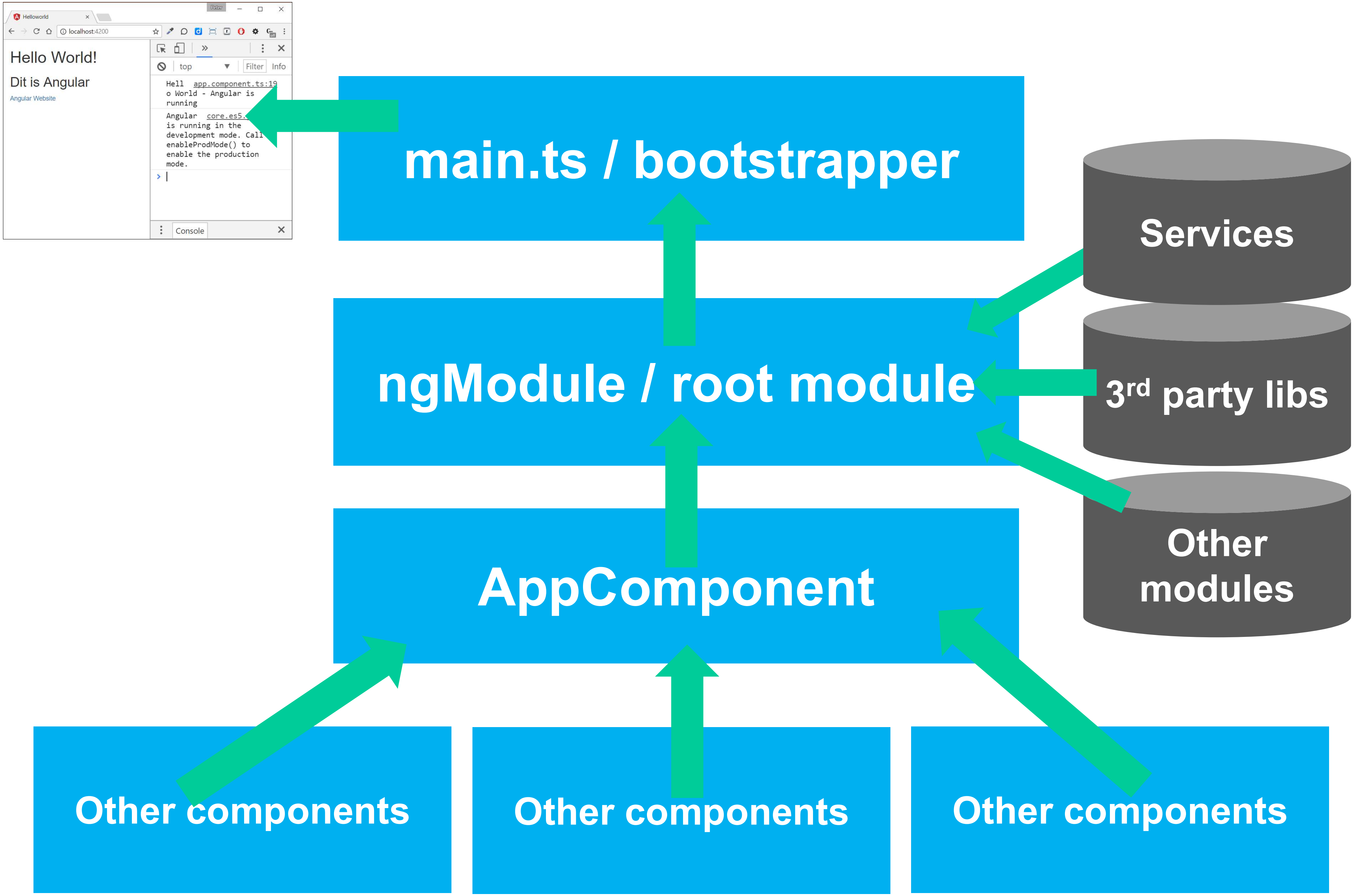
Angular 2 - components



"An Angular-app is a tree of components”

Components – visual representation





Helloworld

localhost:4200

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.
```

main.ts / bootstrapper

ngModule / root module

AppComponent

Services

3rd party libs

Other modules

Other components

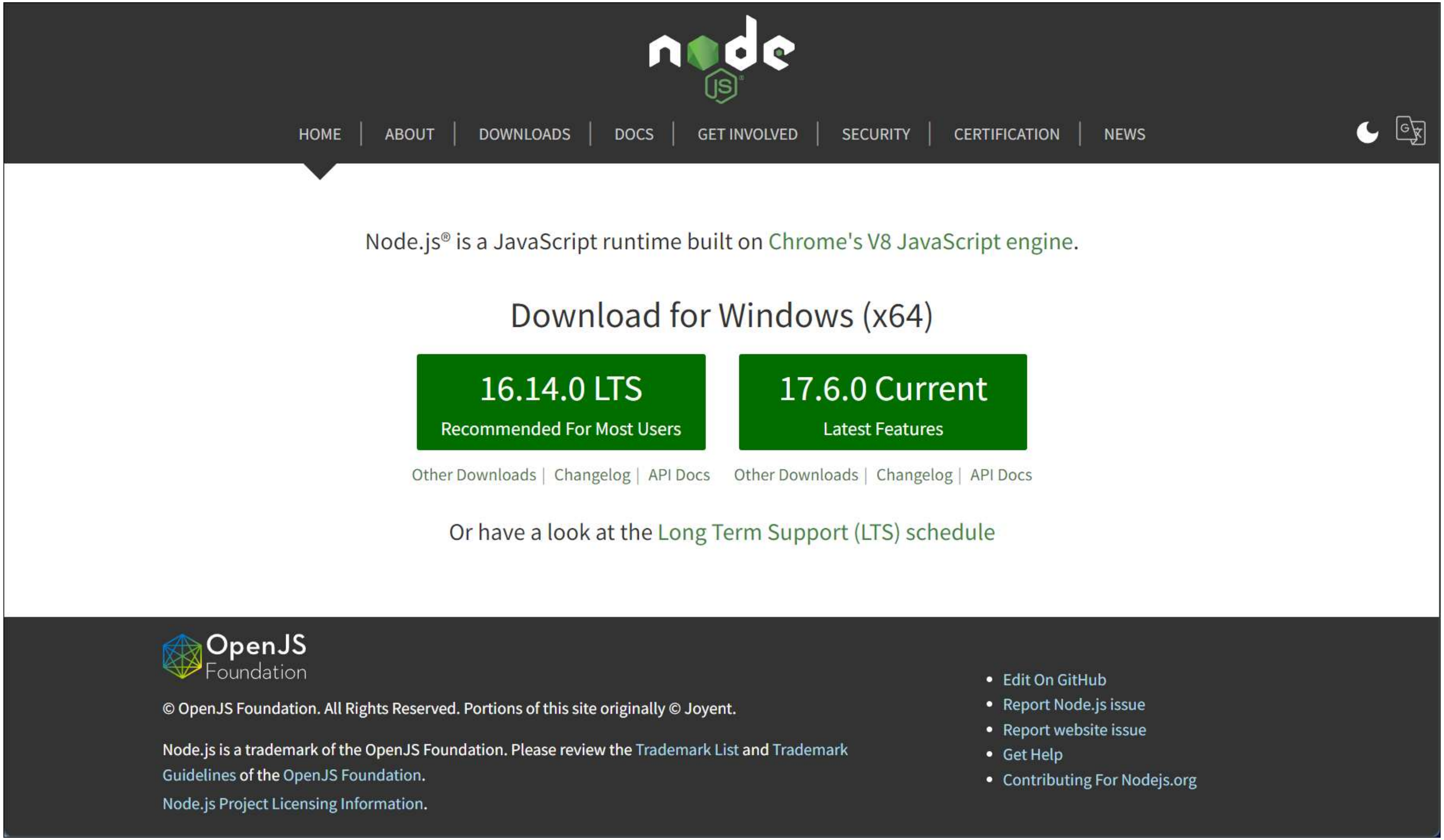
Other components

Other components

Let’s write some code

Hello World in Angular

Angular development dependency: NodeJS 12.20+



HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

16.14.0 LTS
Recommended For Most Users
17.6.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule

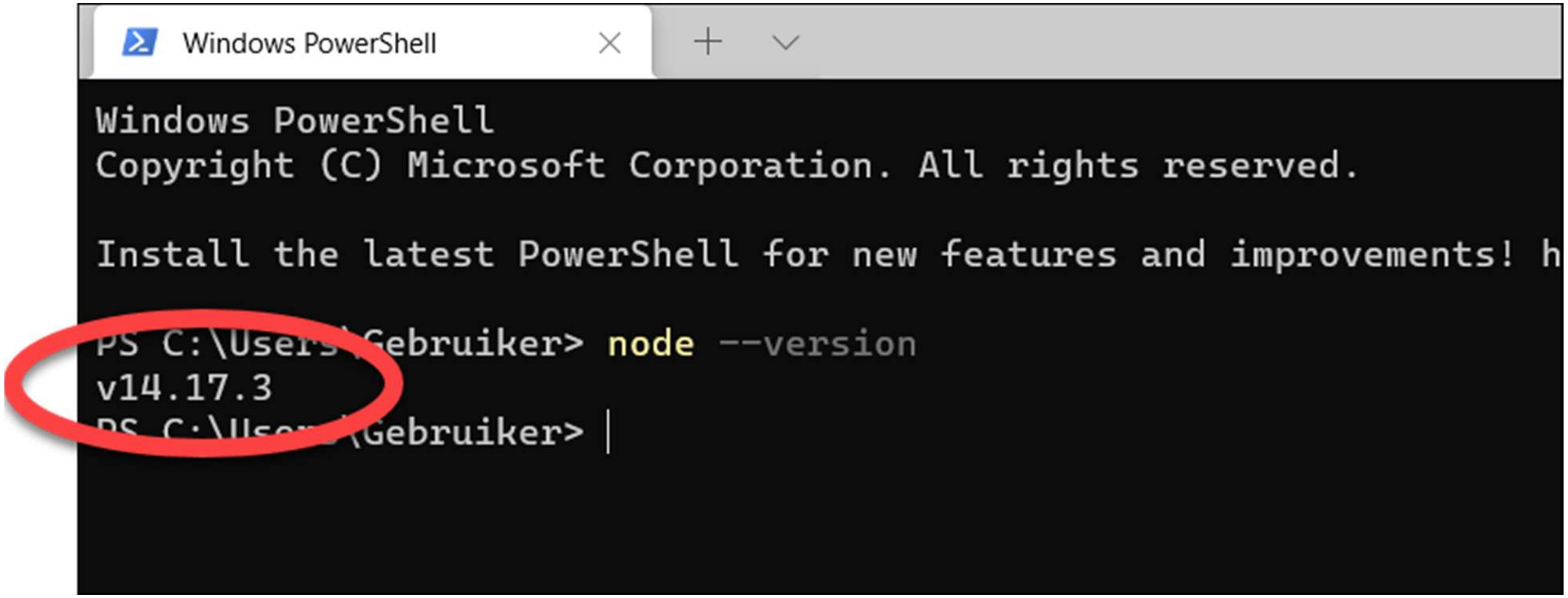
© OpenJS Foundation. All Rights Reserved. Portions of this site originally © Joyent.

Node.js is a trademark of the OpenJS Foundation. Please review the Trademark List and Trademark Guidelines of the OpenJS Foundation.

Node.js Project Licensing Information.

- Edit On GitHub
- Report Node.js issue
- Report website issue
- Get Help
- Contributing For Nodejs.org

Node – check your version

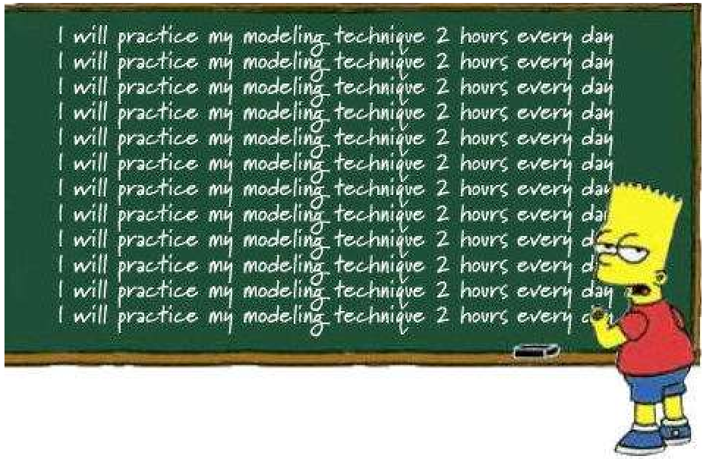


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! h

PS C:\Users\Gebruiker> node --version
v14.17.3
PS C:\Users\Gebruiker>
```

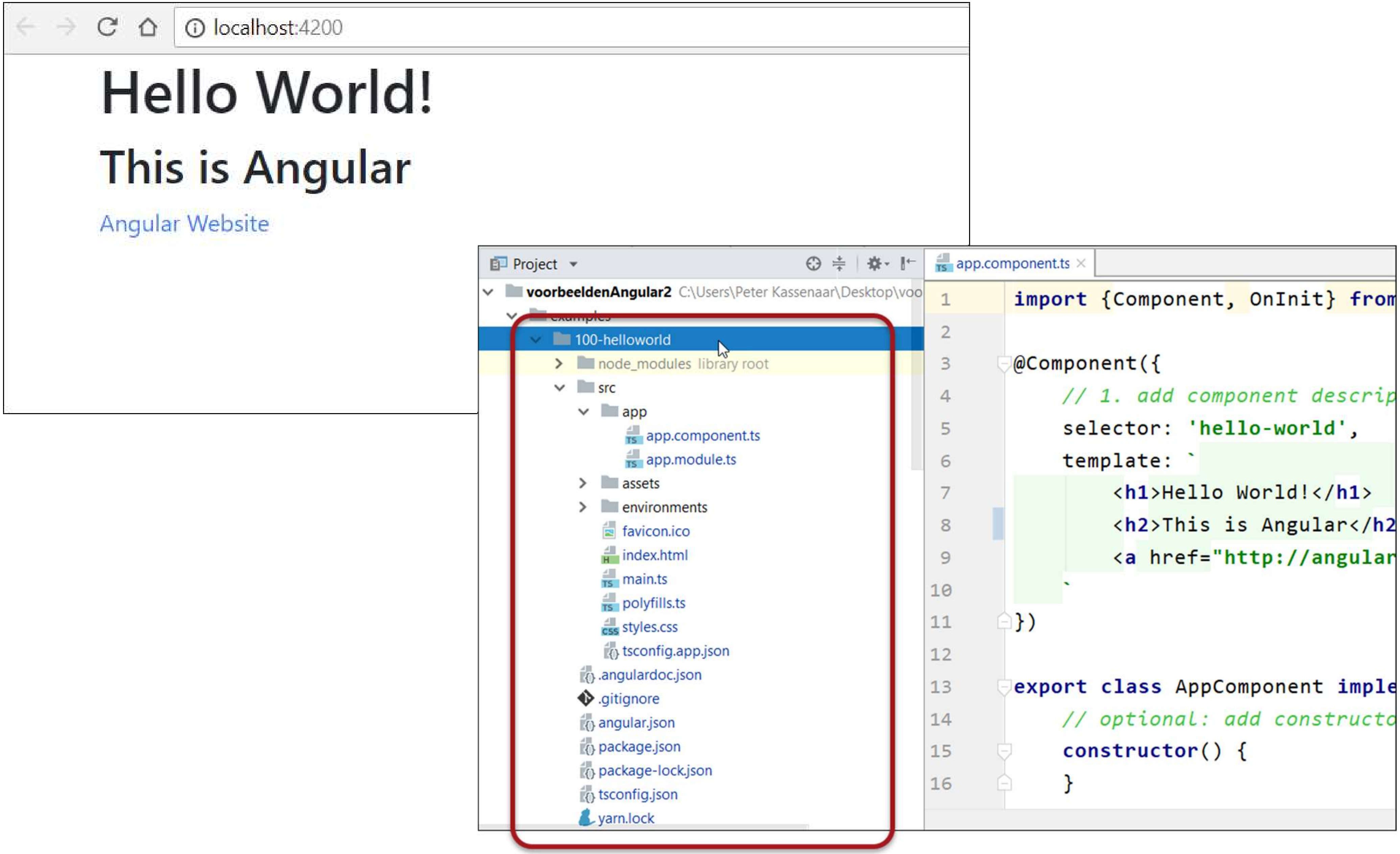
Mini workshop

- Download or clone https://github.com/PeterKassenaar/voorbeeldenAngular2

```
cd examples

cd 100-helloworld

npm install

npm start
```



- Go to browser: http://localhost:4200



Hello World!

This is Angular

Angular Website

```
voorbeeldenAngular2  C:\Users\Peter Kassenaar\Desktop\voo
  examples
    100-helloworld
      node_modules  library root
      src
        app
          app.component.ts
          app.module.ts
        assets
        environments
        favicon.ico
        index.html
        main.ts
        polyfills.ts
        styles.css
        tsconfig.app.json
      .angulardoc.json
      .gitignore
      angular.json
      package.json
      package-lock.json
      tsconfig.json
      yarn.lock
```

```
import {Component, OnInit} from

@Component({
    // 1. add component descrip
    selector: 'hello-world',
    template: `
        <h1>Hello World!</h1>
        <h2>This is Angular</h2
        <a href="http://angular
    `
})

export class AppComponent imple
    // optional: add constructo
    constructor() {
    }
```

Boilerplate code for Hello World

Steps

1. Set up environment, boilerplate & libraries
   - Important configuration files
2. Angular Component(s)
3. Angular Module(s): @ngModule()
4. Bootstrap our module
5. Write HTML-pagina (`index.html`)

Boilerplate files #1/3 - package.json

```
{
  "name": "hello-angular",
  "description": "Voorbeeldproject bij de training Angular (C) - info@kassenaar.com",
  "version": "0.0.1",
  "license": "MIT",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "6.0.0",
    "@angular/common": "6.0.0",
    "@angular/compiler": "6.0.0",
    "@angular/core": "6.0.0",
    "@angular/forms": "6.0.0",
    "rxjs": "^6.1.0",
    "zone.js": "^0.8.26"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.6.0",
    "@angular/cli": "6.0.0",
    "typescript": "2.7.2"
  },
  "author": "Peter Kassenaar <info@kassenaar.com>"
}
```

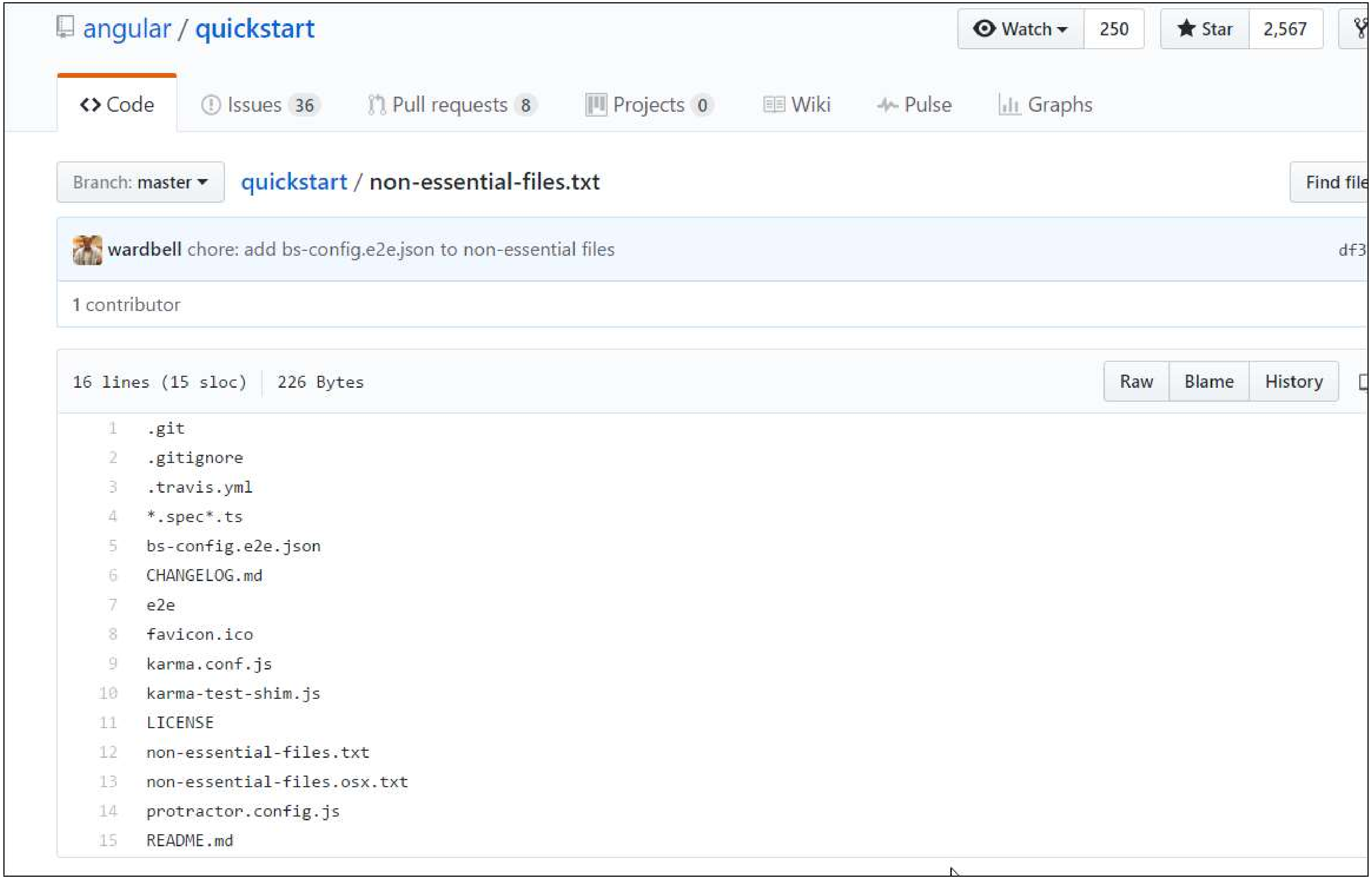
Boilerplate files #2/3 - **tsconfig.json**

```
{
  "compileOnSave"  : false,
  "compilerOptions": {
   "outDir"                 : "./dist/out-tsc",
   "baseUrl"                : "src",
   "sourceMap"              : true,
   "declaration"            : false,
   "moduleResolution"       : "node",
   "emitDecoratorMetadata"  : true,
   "experimentalDecorators": true,
   "target"                 : "es5",
   "typeRoots"              : [
     "node_modules/@types"
   ],
   "lib"                    : [
     "es2016",
     "dom"
   ]
  }
}
```

Boilerplate files #3/3 – **angular.json**

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "helloworld": {
      "root": "",
      "sourceRoot": "src",
      "projectType": "application",
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "outputPath": "dist",
            "index": "src/index.html",
            "main": "src/main.ts",
            "tsConfig": "src/tsconfig.app.json",
      …
}
```

"Nice to have" - non-essential files



```
.git
.gitignore
.travis.yml
*.spec*.ts
bs-config.e2e.json
CHANGELOG.md
e2e
favicon.ico
karma.conf.js
karma-test-shim.js
LICENSE
non-essential-files.txt
non-essential-files.osx.txt
protractor.config.js
README.md
```

https://github.com/angular/quickstart/blob/master/non-essential-files.txt

Step 2 – Component

Convention - components in directory `/src/app`

Or: edit in `angular.json`

Filename: `src/app/app.component.ts`

```
import {Component} from ‘@angular/core';
@Component({
   selector: 'hello-world',
   template: '<h1>Hello Angular</h1>'
})
export class AppComponent {

}
```

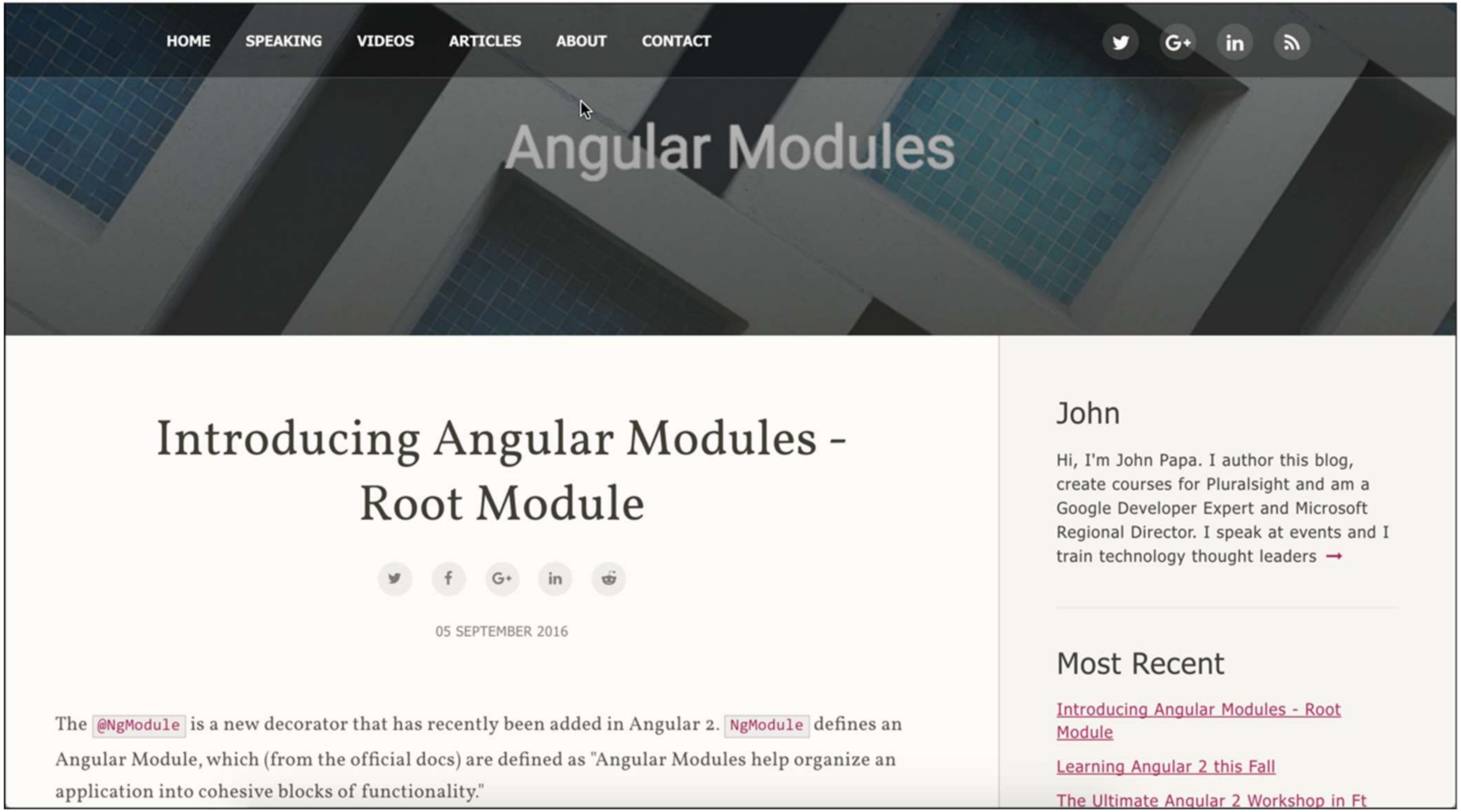
Step 3 – @ngModule

Convention - filename: `/src/app.module.ts`

```
// Angular Modules
import {NgModule}      from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';

// Custom Components
import {AppComponent} from './app.component';

// Module declaration
@NgModule({
   imports     : [BrowserModule],
   declarations: [AppComponent],
   bootstrap   : [AppComponent]
})
export class AppModule {
}
```

Root Module of the application

Some background info on Root Module



Introducing Angular Modules - Root Module

05 SEPTEMBER 2016

The `@NgModule` is a new decorator that has recently been added in Angular 2. `NgModule` defines an Angular Module, which (from the official docs) are defined as "Angular Modules help organize an application into cohesive blocks of functionality."

John

Hi, I'm John Papa. I author this blog, create courses for Pluralsight and am a Google Developer Expert and Microsoft Regional Director. I speak at events and I train technology thought leaders →

Most Recent

Introducing Angular Modules - Root Module

Learning Angular 2 this Fall

The Ultimate Angular 2 Workshop in Ft

https://johnpapa.net/introducing-angular-modules-root-module/

Step 4 - bootstrap component

Best practice: bootstrap app in separate component

Convention: `main.ts`, of `app.main.ts`.

```
import {enableProdMode} from '@angular/core';
import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';

import {AppModule} from './app/app.module';
import {environment} from './environments/environment';

if (environment.production) {
   enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule);
```

Step 5 – index.html

`index.html` - simple HTML file - expanded at runtime by WebPack

Header:

```
<html>


<head>
  <meta charset="utf-8">
  <title>Helloworld</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
```

Body van index.html

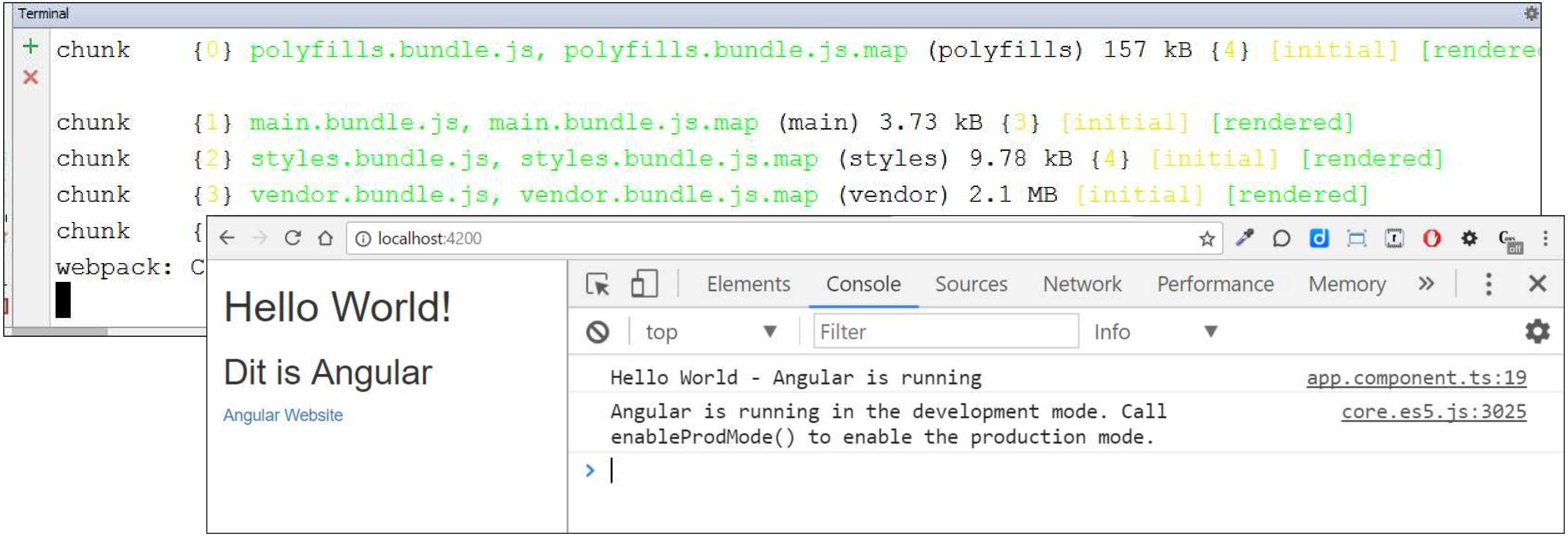
Verwijzing naar de root-component:

```
<body>
  <hello-world>
    Bezig met laden...
  </hello-world>
</body>
```

App draaien

`npm start` – draait de scriptopdracht start uit `package.json`.

`ng serve` - start globale angular-cli instantie



```
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 157 kB {4} [initial] [rendere

chunk    {1} main.bundle.js, main.bundle.js.map (main) 3.73 kB {3} [initial] [rendered]
chunk    {2} styles.bundle.js, styles.bundle.js.map (styles) 9.78 kB {4} [initial] [rendered]
chunk    {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.1 MB [initial] [rendered]
chunk    {
webpack: C
```

Hello World!

Dit is Angular

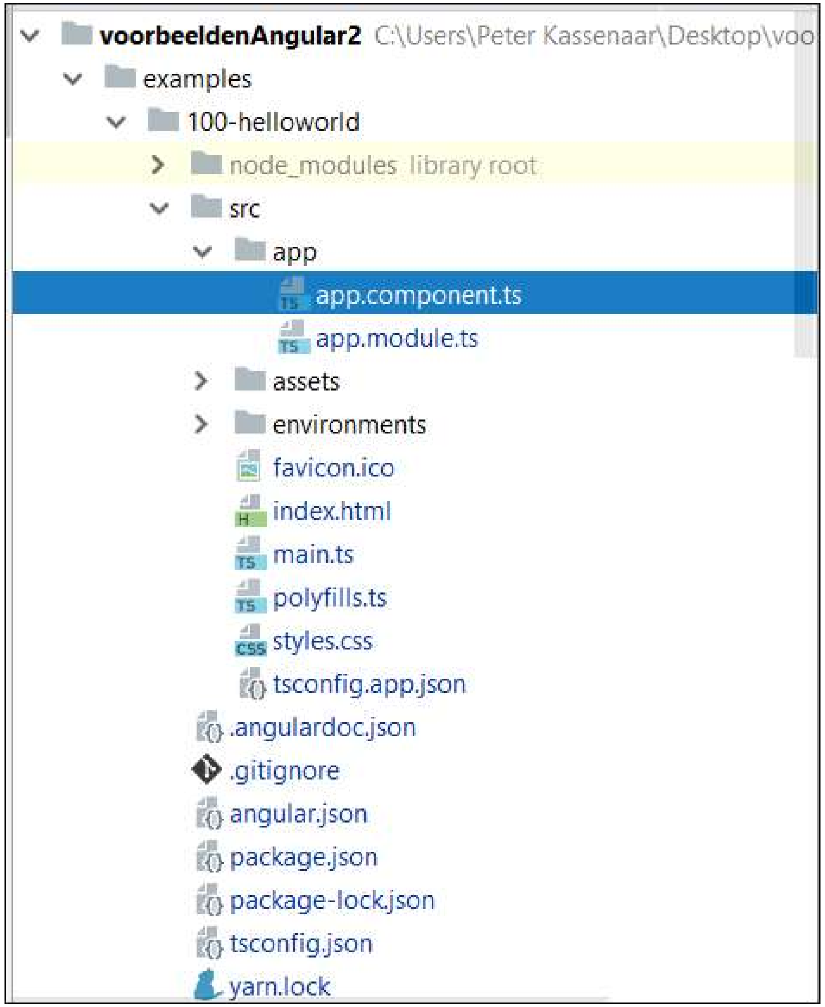
Angular Website

```
Hello World - Angular is running                                    app.component.ts:19
Angular is running in the development mode. Call                   core.es5.js:3025
enableProdMode() to enable the production mode.
>
```

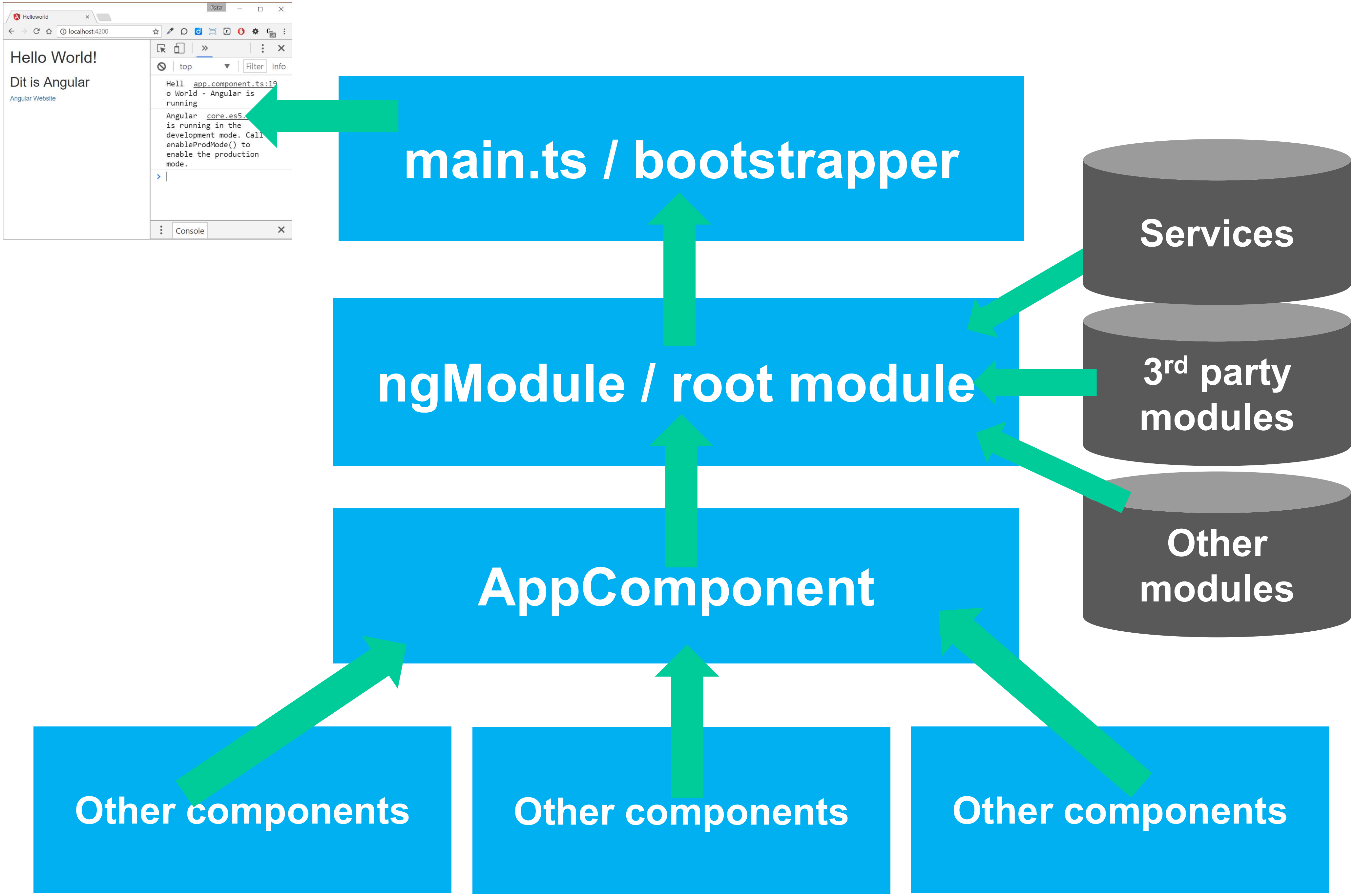
Daarna: wijzigingen aanbrengen in `app.component.ts`

– worden opgepikt door Live Reload

Basic Project Structure



```
voorbeeldenAngular2 C:\Users\Peter Kassenaar\Desktop\voo
  examples
    100-helloworld
      node_modules library root
      src
        app
          app.component.ts
          app.module.ts
        assets
        environments
        favicon.ico
        index.html
        main.ts
        polyfills.ts
        styles.css
        tsconfig.app.json
      .angulardoc.json
      .gitignore
      angular.json
      package.json
      package-lock.json
      tsconfig.json
      yarn.lock
```



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.
>
```

Assets

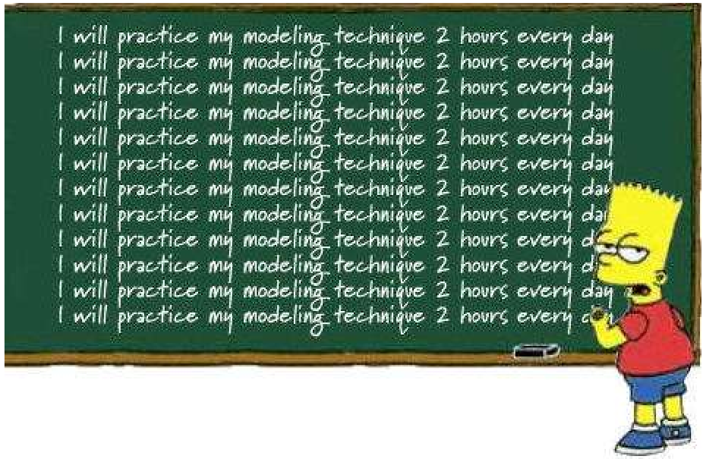
github.com/PeterKassenaar/voorbeeldenAngular2

Oefeningen en meer voorbeeldcode

Checkpoint

- Er is aardig wat boilerplate code nodig om een Angular-app te starten
- Vier stappen
  1. Set up environment, boilerplate & libraries
  2. Schrijf Angular Root Component voor de app
  3. Bootstrap de component
  4. Schrijf HTML-pagina (`index.html`)
- Daarna: app gaan uitbreiden
- Oefening `1a), 1b), 1c), 1d)`

Workshop....



Angular CLI

Snel nieuwe projecten instellen via de command line

Angular-CLI to the rescue

- Het *is* mogelijk nieuwe Angular-projecten from scratch te starten.
- Met de CLI is eenvoudiger.
- CLI-options:
  - Scaffolding
  - Generating
  - Testing
  - Building
  - AOT-Compiling
  - …

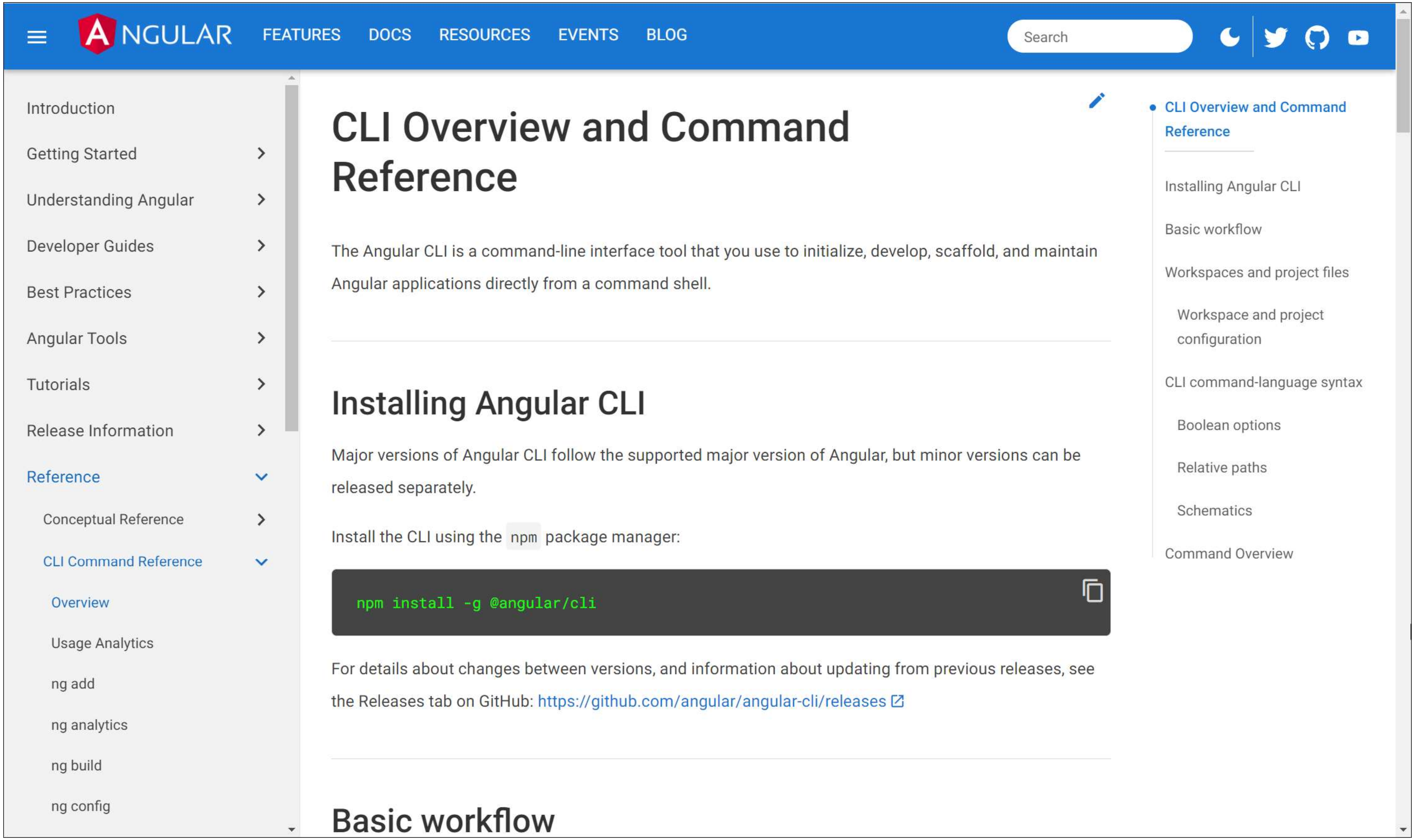
Scaffolding - Angular CLI

Projecten, componenten, routes en meer definiëren vanaf de command line

https://github.com/angular/angular-cli

en

https://cli.angular.io/



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:

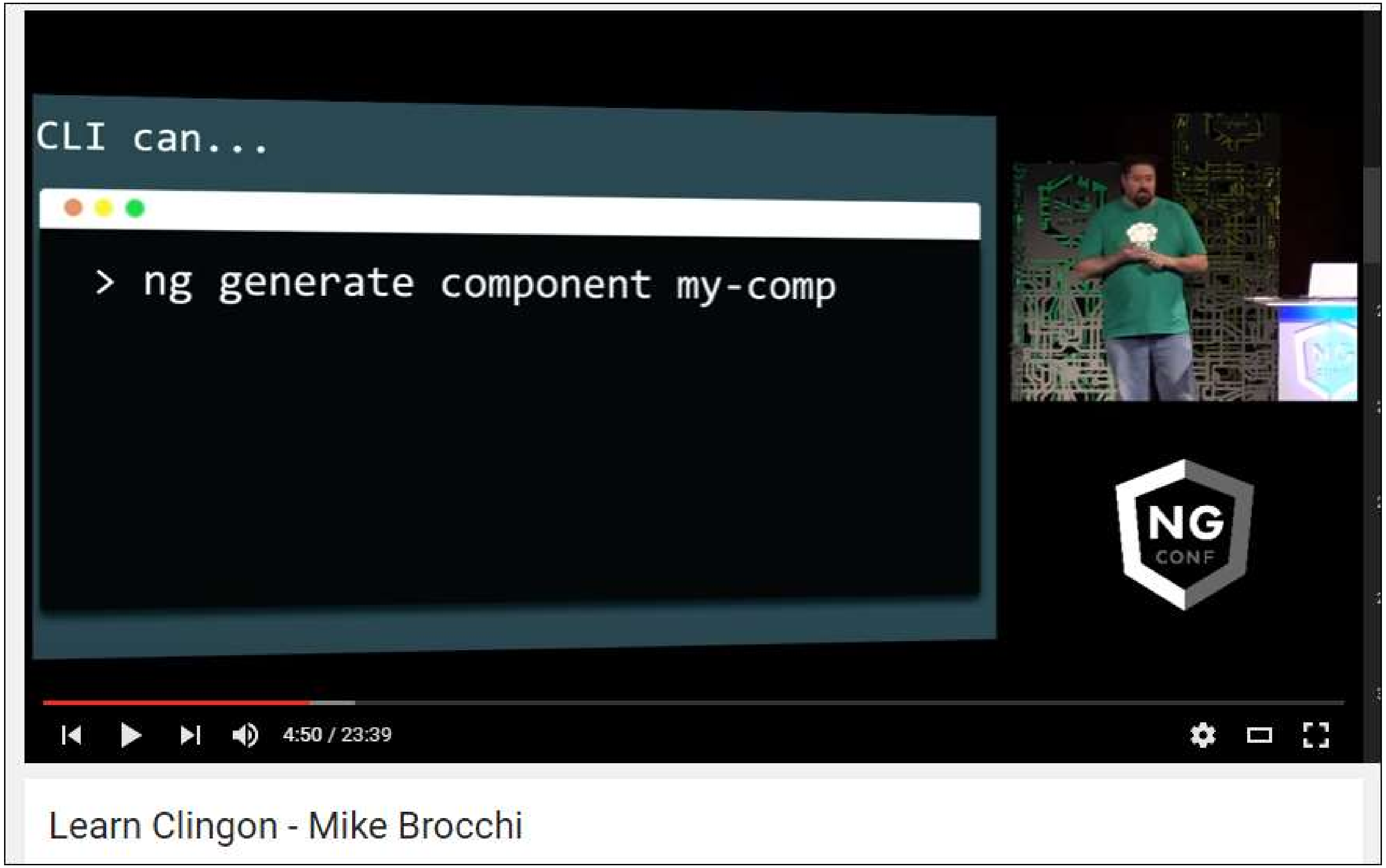
```
npm install -g @angular/cli
```

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

Basic workflow

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

Achtergrondinformatie – Mike Brocchi



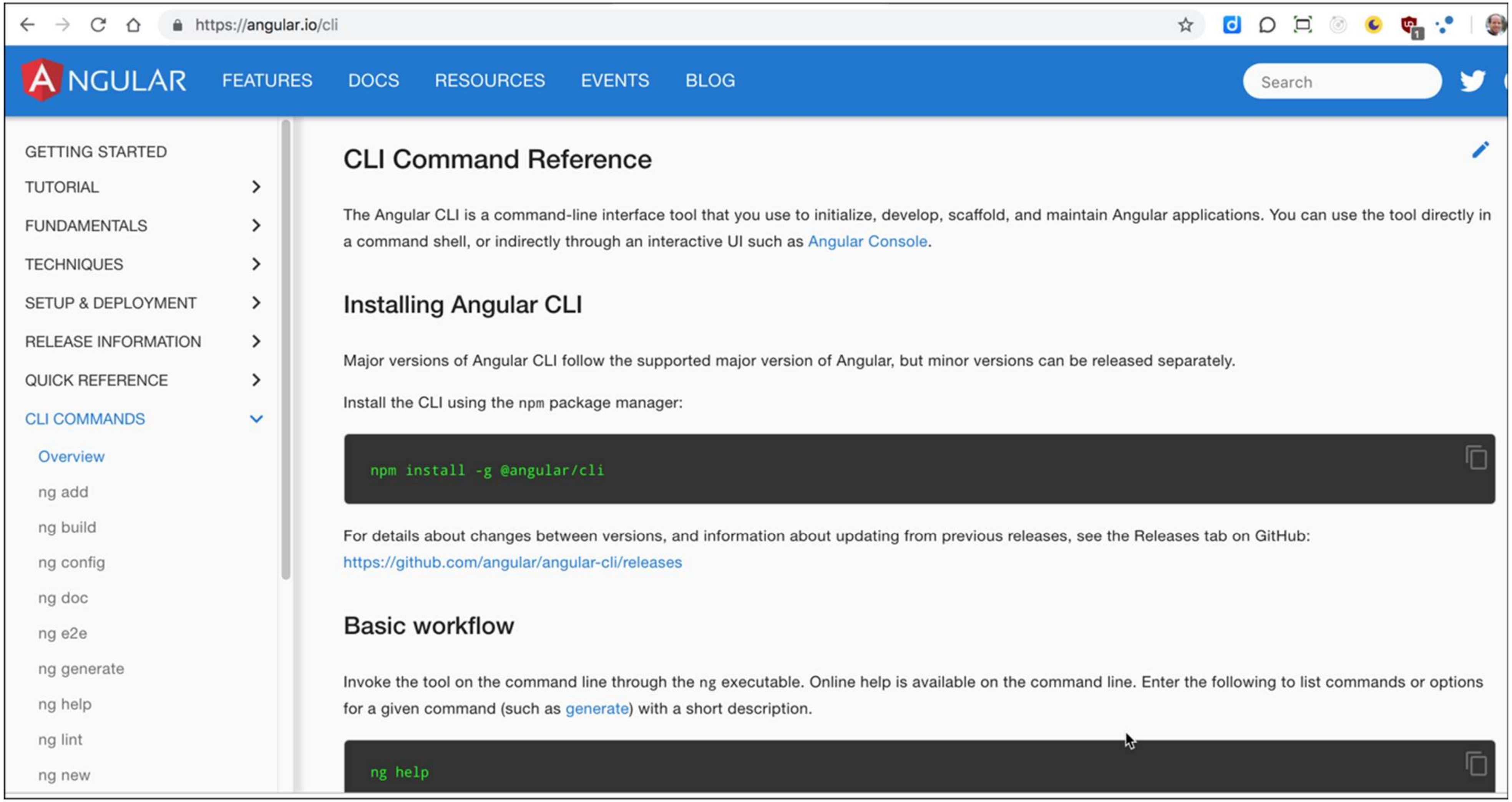
CLI can...

```
> ng generate component my-comp
```

Learn Clingon - Mike Brocchi

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

Documentatie - in de Angular Docs



CLI Command Reference

The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications. You can use the tool directly in a command shell, or indirectly through an interactive UI such as Angular Console.

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

Basic workflow

Invoke the tool on the command line through the ng executable. Online help is available on the command line. Enter the following to list commands or options for a given command (such as generate) with a short description.

```
ng help
```

https://angular.io/cli

Angular 2 Code - Backend

Kort over TypeScript en ES6

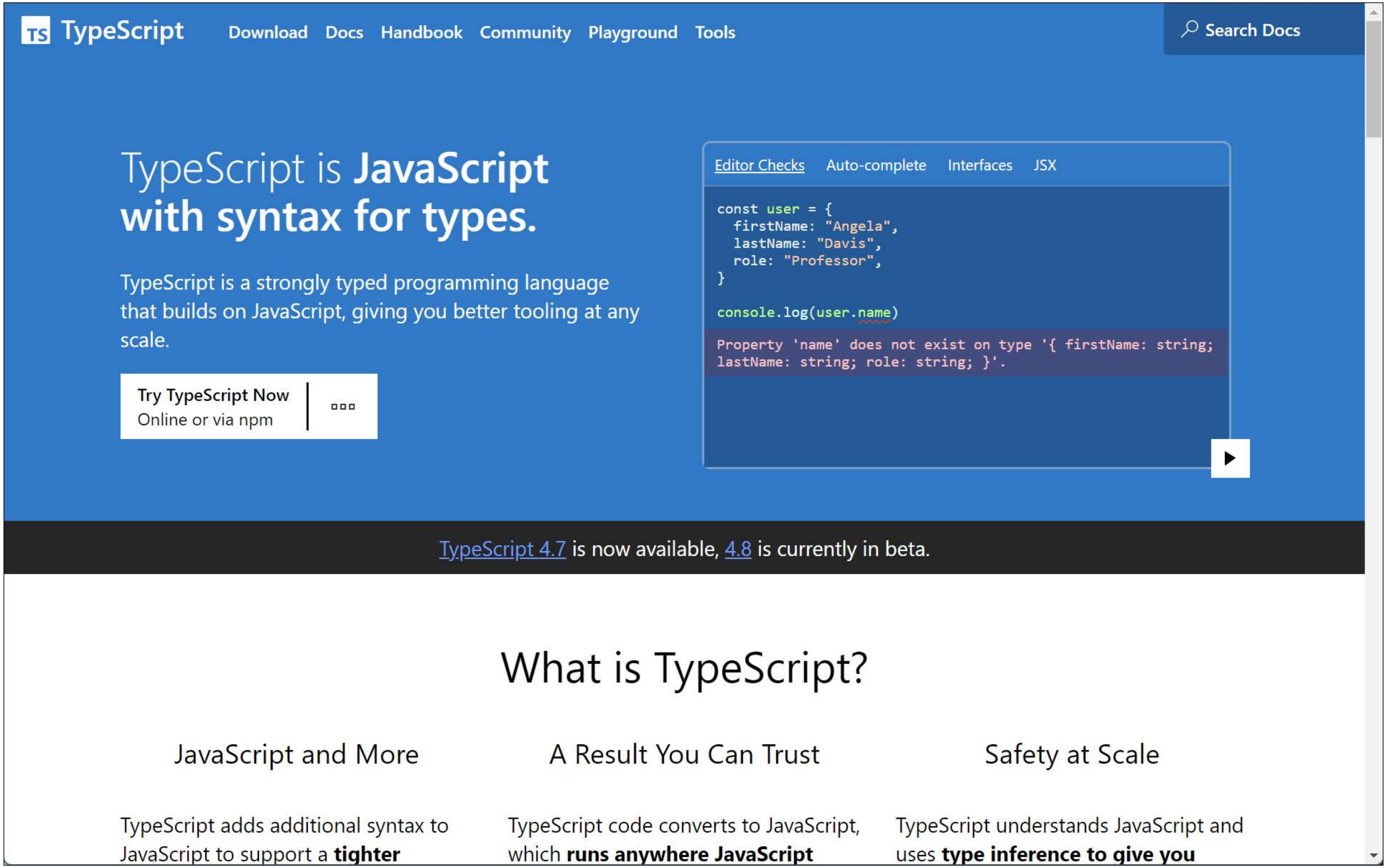
Programmeertalen

TypeScript

- Beperkt in deze module
- Voor zover nodig om Angular te gebruiken.
- Maar, TypeScript is niet beperkt tot Angular!
- Algemene superset van JavaScript
- Compileert plain JavaScript, dat browsers begrijpen
- Twee taken

    1. Compiling/Transpiling
    2. Type Safety

TypeScript website



TypeScript is **JavaScript with syntax for types.**

TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale.

```
const user = {
  firstName: "Angela",
  lastName: "Davis",
  role: "Professor",
}

console.log(user.name)
```

```
Property 'name' does not exist on type '{ firstName: string;
lastName: string; role: string; }'.
```

TypeScript 4.7 is now available, 4.8 is currently in beta.

What is TypeScript?

JavaScript and More

TypeScript adds additional syntax to JavaScript to support a **tighter**

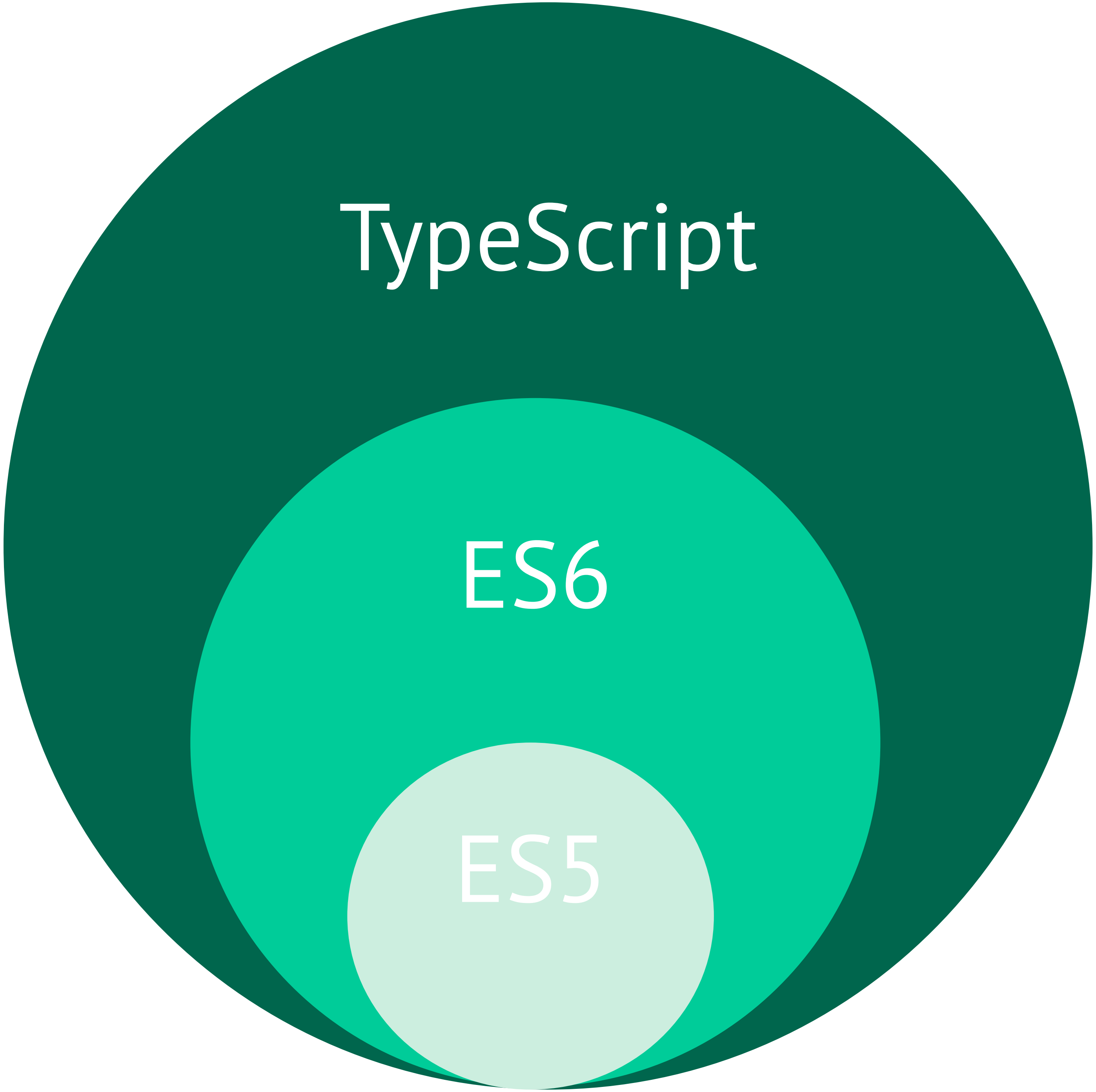
A Result You Can Trust

TypeScript code converts to JavaScript, which **runs anywhere JavaScript**

Safety at Scale

TypeScript understands JavaScript and uses **type inference to give you**

www.typescriptlang.org


TypeScript
ES6
ES5

ES6 en TypeScript

De toekomst van JavaScript is ES6/ES2015

- Major update van JavaScript als programmeertaal
- Modules, classes en meer
- Verplicht in Angular, optioneel in JavaScript, Vue, React, …

TypeScript breidt ES6 verder uit

- Annotaties & types
- Interfaces
- Compiler,
- …

TypeScript – tooling support

Types, Autocompletion.

Compile-time checking in editors.

Alles is optioneel. Je kunt altijd nog gewoon JavaScript gebruiken.

Onderdelen van een Component Class

imports

```
import { Component } from ’@angular/core';
import { DataService } from './services/data-service';
```

annotations

```
@Component({
  selector: 'orders',
  directives: [DataService],
  templateUrl: 'orders-component.html',
})
```

class

```
export class OrdersComponent {
          …
}
```

Checkpoint

- Angular 2 is een totaal ander framework dan Angular 1
- Component-based vs. Page-based
- Nieuwe syntaxis
- Nieuwe programmeertalen en andere nieuwe kenmerken
- Concepten komen – grotendeels – overeen
- Veel boilerplate-code nodig voor een Quickstart
- Daarna: niet meer naar omkijken. Concentreren op de componenten