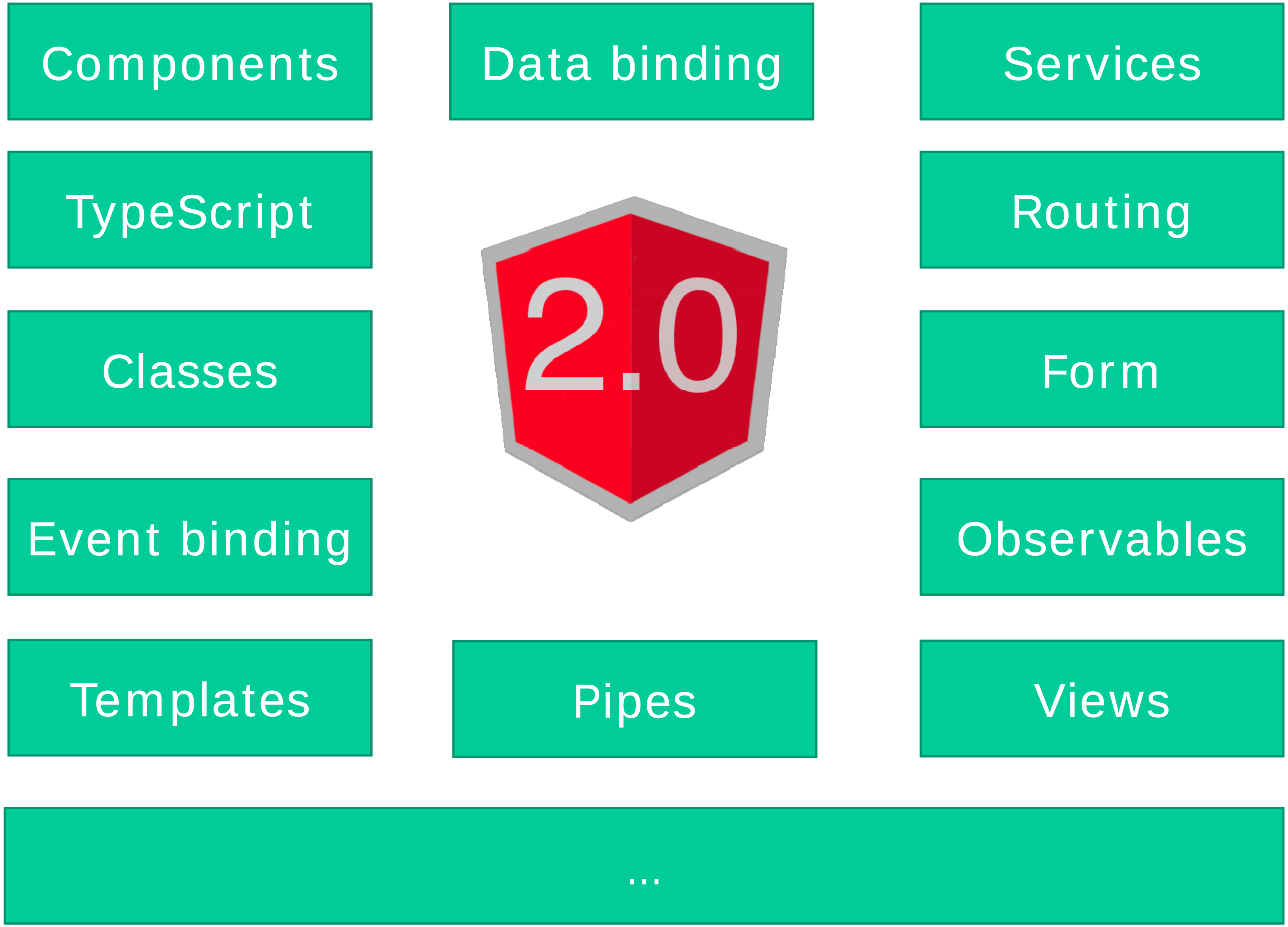
Angular 2 – dag 2
Korte herhaling

Peter Kassenaar –
info@kassenaar.com

Framework to Platform

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

Angular 2



Components

Data binding

Services

TypeScript

Routing

Classes

Form

Event binding

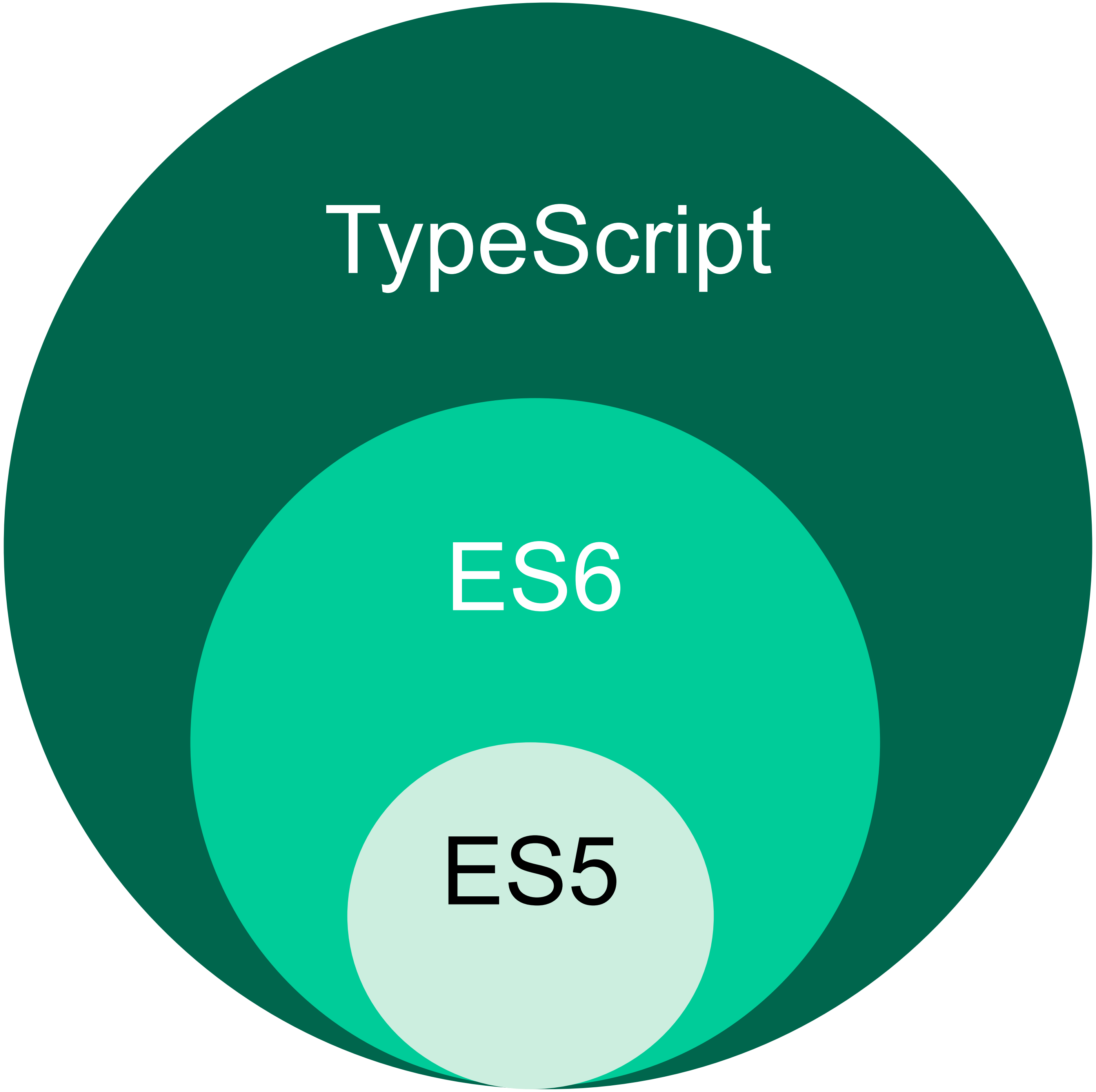
Observables

Templates

Pipes

Views

...


TypeScript
ES6
ES5

Boilerplate code - stappenplan

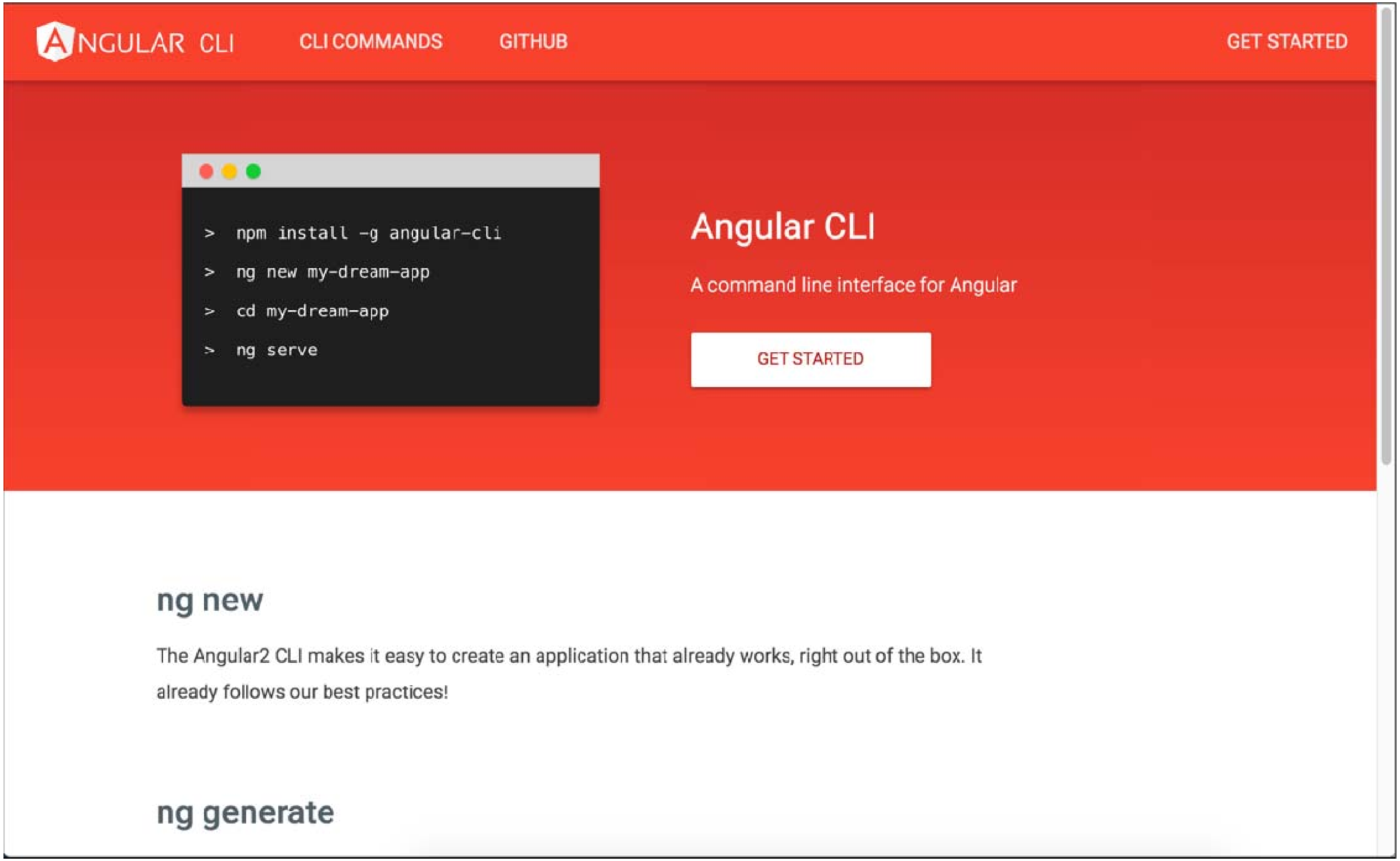
1. Set up environment, boilerplate & libraries
   - `npm install`
2. Schrijf Angular Root Component & Module voor de app
3. Bootstrap
4. Schrijf HTML-pagina (`index.html`)
5. Test applicatie
   - `npm start`

Data binding

- Nieuwe notatiewijzen in HTML-views/partials.

    1. Simple data binding **{{ … }}**
    2. Event binding **(…)="..."**
    3. One-way data binding **[…]="..."**
    4. Two-way data binding **[(ngModel)] = " …"**

Angular CLI



```
> 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

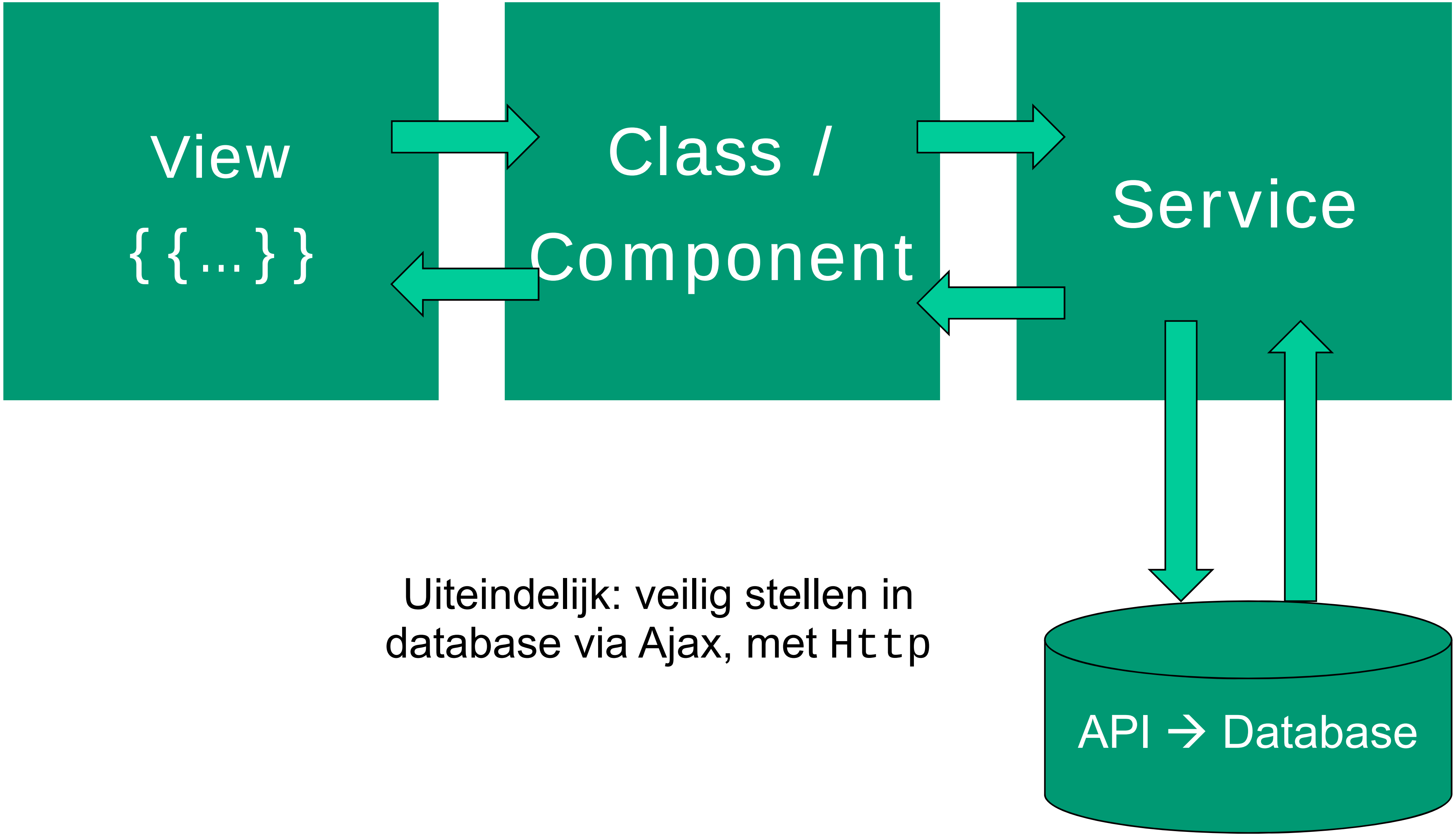
The Angular2 CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

ng generate

Gisteren - vervolg

- Services
  - Static services
  - Observables / rxJS
  - GET-ing data

Data flow



View

{{…}}

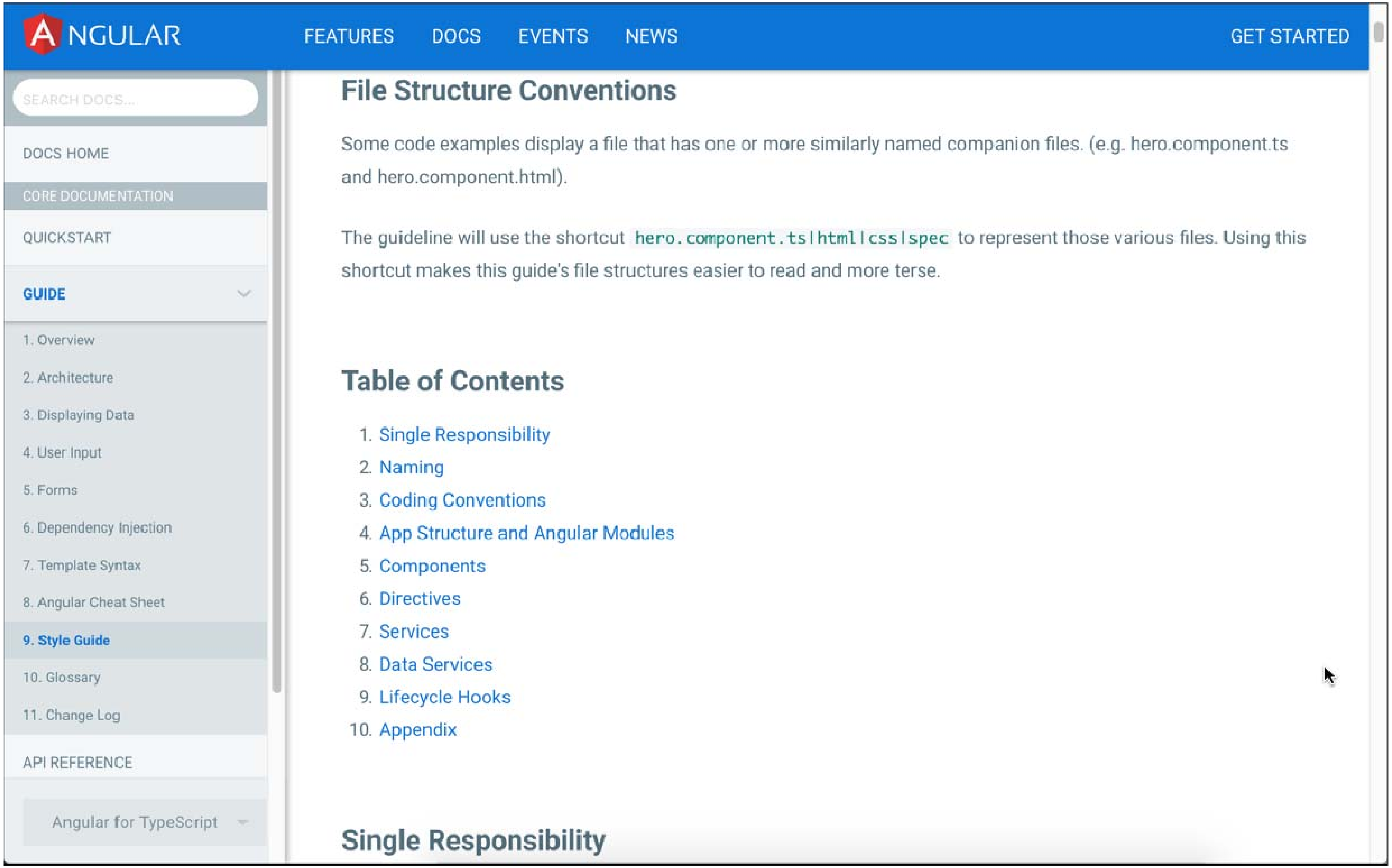
Class / Component

Service

API → Database

Uiteindelijk: veilig stellen in database via Ajax, met `Http`

Style Guide - conventions



File Structure Conventions

Some code examples display a file that has one or more similarly named companion files. (e.g. hero.component.ts and hero.component.html).

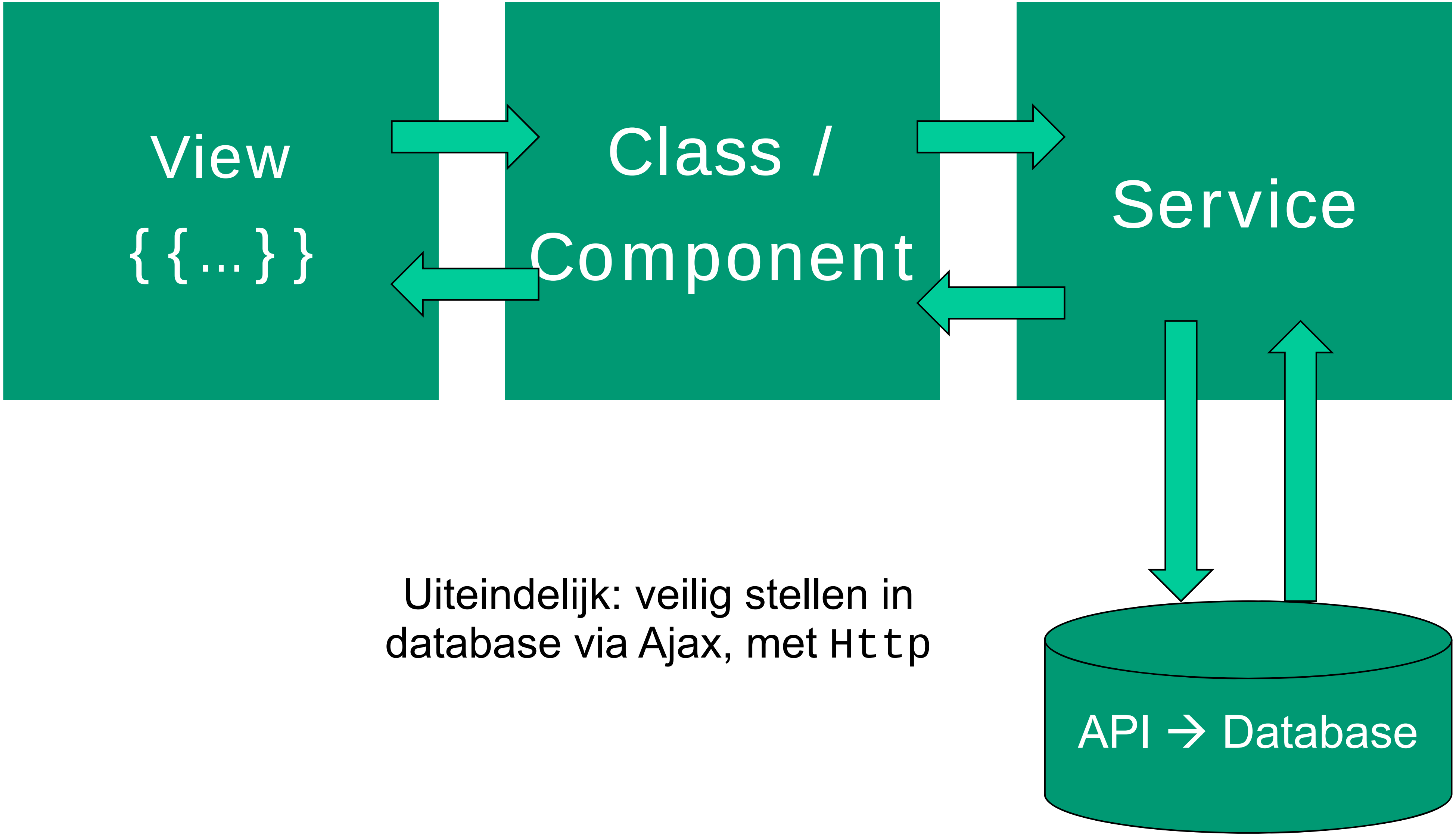
The guideline will use the shortcut `hero.component.ts|html|css|spec` to represent those various files. Using this shortcut makes this guide's file structures easier to read and more terse.

Table of Contents

Single Responsibility

https://angular.io/docs/ts/latest/guide/style-guide.html

Service Data Flow



View

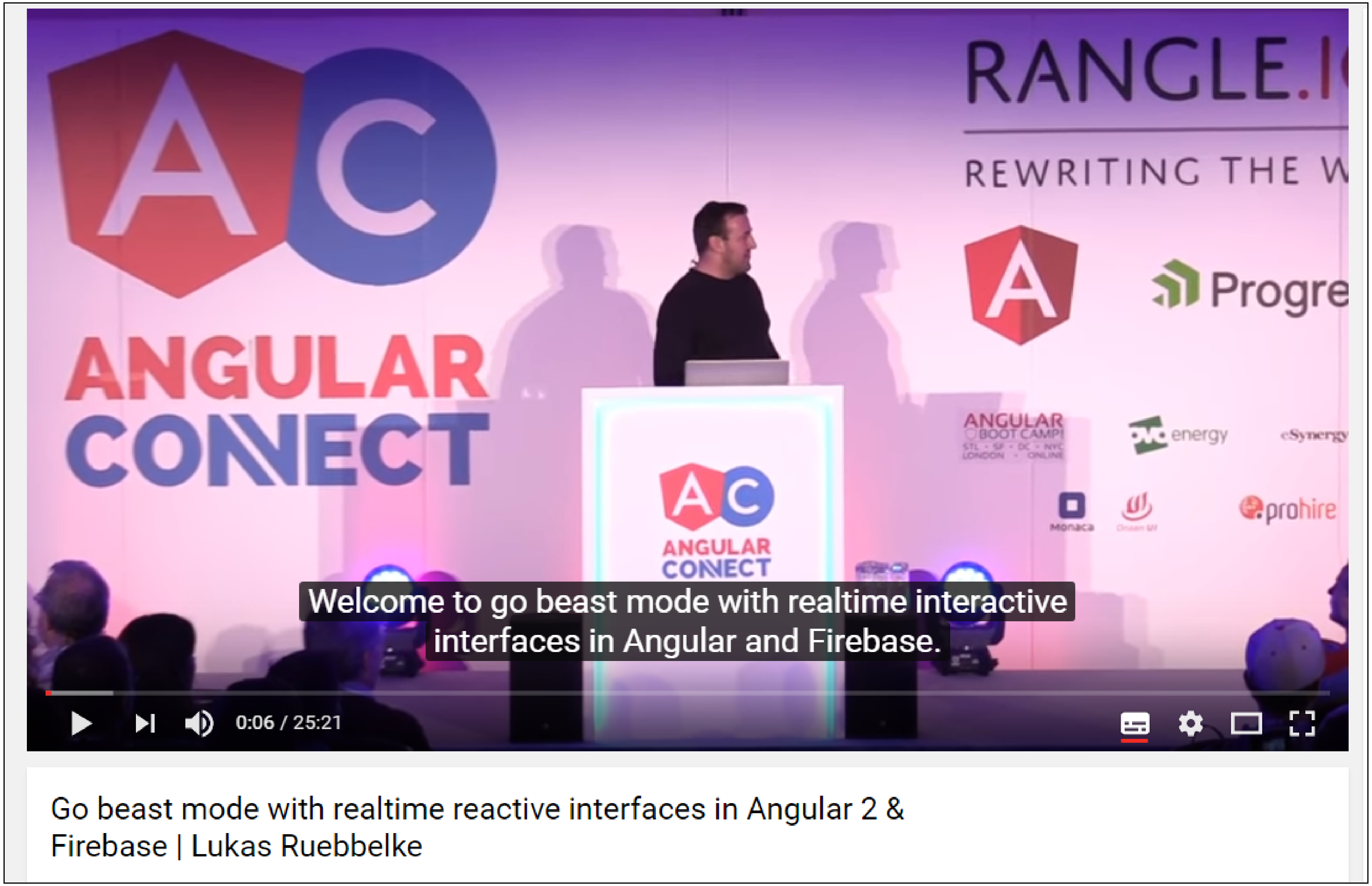
{{…}}

Class / Component

Service

API → Database

Uiteindelijk: veilig stellen in database via Ajax, met `Http`

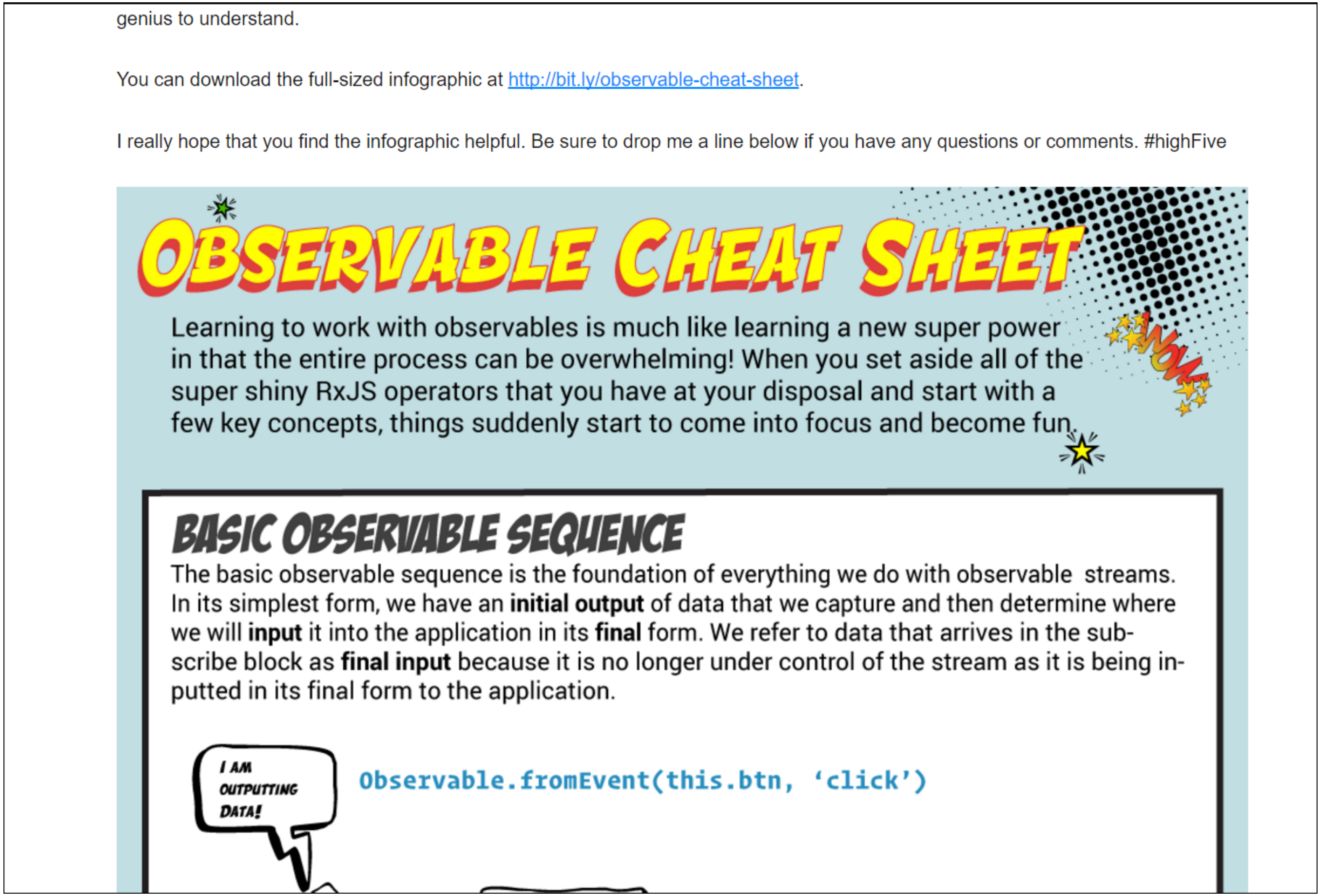


Welcome to go beast mode with realtime interactive interfaces in Angular and Firebase.

Go beast mode with realtime reactive interfaces in Angular 2 & Firebase | Lukas Ruebbelke

https://www.youtube.com/watch?v=5CTL7aqSvJU

Observable Cheat Sheet



genius to understand.

You can download the full-sized infographic at http://bit.ly/observable-cheat-sheet.

I really hope that you find the infographic helpful. Be sure to drop me a line below if you have any questions or comments. #highFive

OBSERVABLE CHEAT SHEET

Learning to work with observables is much like learning a new super power in that the entire process can be overwhelming! When you set aside all of the super shiny RxJS operators that you have at your disposal and start with a few key concepts, things suddenly start to come into focus and become fun.

BASIC OBSERVABLE SEQUENCE

The basic observable sequence is the foundation of everything we do with observable streams. In its simplest form, we have an **initial output** of data that we capture and then determine where we will **input** it into the application in its **final** form. We refer to data that arrives in the subscribe block as **final input** because it is no longer under control of the stream as it is being inputted in its final form to the application.

```
Observable.fromEvent(this.btn, ‘click’)
```

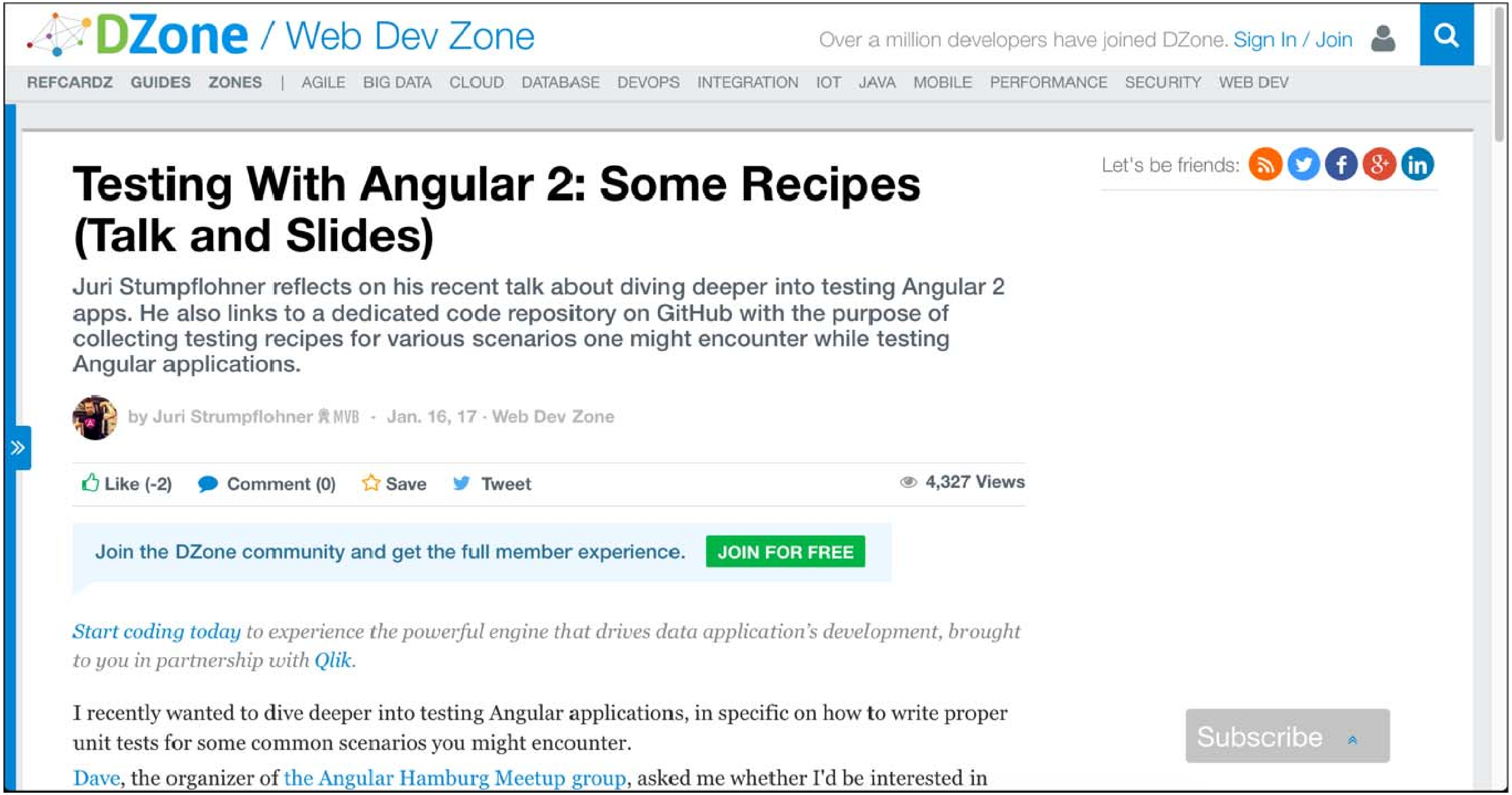
http://onehungrymind.com/observable-cheat-sheet/

Vandaag

- Extra – Testing / ˜Augury
- Vervolg Observables
    - Live API’s
    - Subscribing to streams
- Tree of components
    - Applicaties opbouwen met meerdere componenten
    - Data flow tussen de componenten
- Custom events en Custom attributes schrijven
- Routing
- (forms)

Testing - some shortcuts

Goed introductie artikel



DZone / Web Dev Zone
Over a million developers have joined DZone. Sign In / Join
REFCARDZ GUIDES ZONES | AGILE BIG DATA CLOUD DATABASE DEVOPS INTEGRATION IOT JAVA MOBILE PERFORMANCE SECURITY WEB DEV
Let's be friends:
Like (-2)
Comment (0)
Save
Tweet
4,327 Views
JOIN FOR FREE
Subscribe

Testing With Angular 2: Some Recipes (Talk and Slides)

Juri Stumpflohner reflects on his recent talk about diving deeper into testing Angular 2 apps. He also links to a dedicated code repository on GitHub with the purpose of collecting testing recipes for various scenarios one might encounter while testing Angular applications.

by Juri Strumpflohner MVB · Jan. 16, 17 · Web Dev Zone

Join the DZone community and get the full member experience.

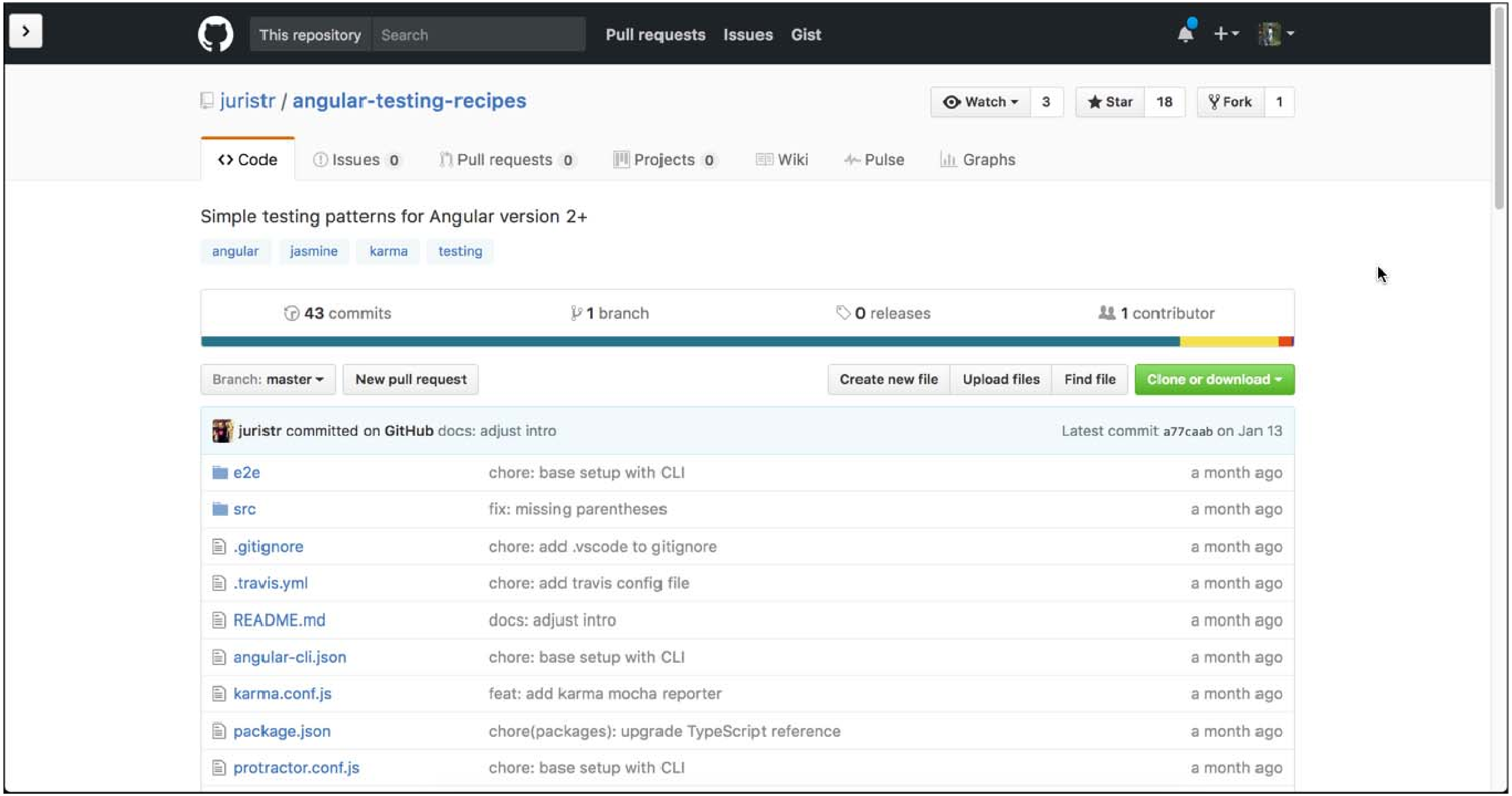
Start coding today to experience the powerful engine that drives data application's development, brought to you in partnership with Qlik.

I recently wanted to dive deeper into testing Angular applications, in specific on how to write proper unit tests for some common scenarios you might encounter.

Dave, the organizer of the Angular Hamburg Meetup group, asked me whether I'd be interested in

https://dzone.com/articles/talk-testing-with-angular-some-recipes

Repo – Angular Testing recipes



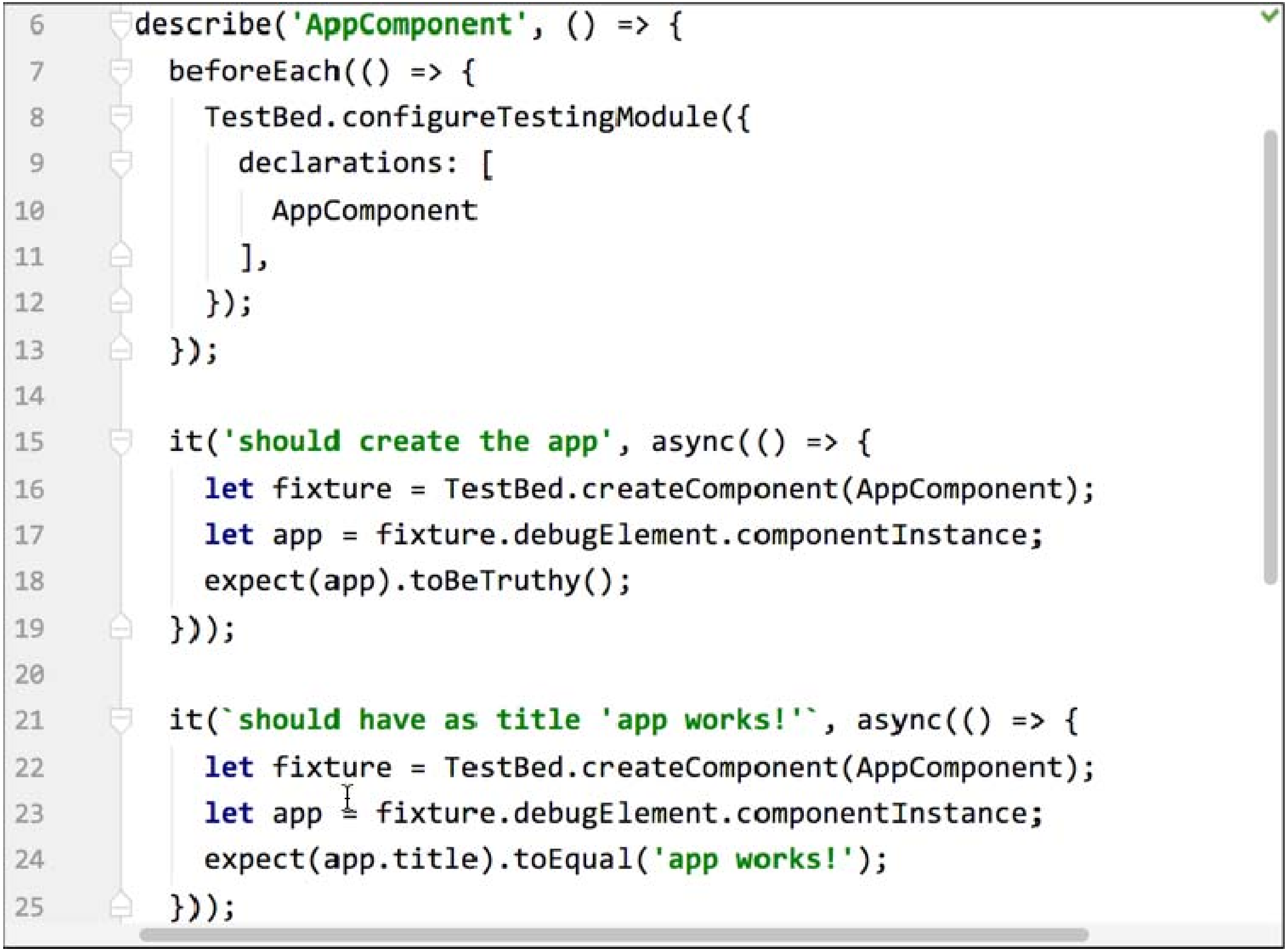
juristr / angular-testing-recipes

Simple testing patterns for Angular version 2+

juristr committed on GitHub docs: adjust intro — Latest commit a77caab on Jan 13

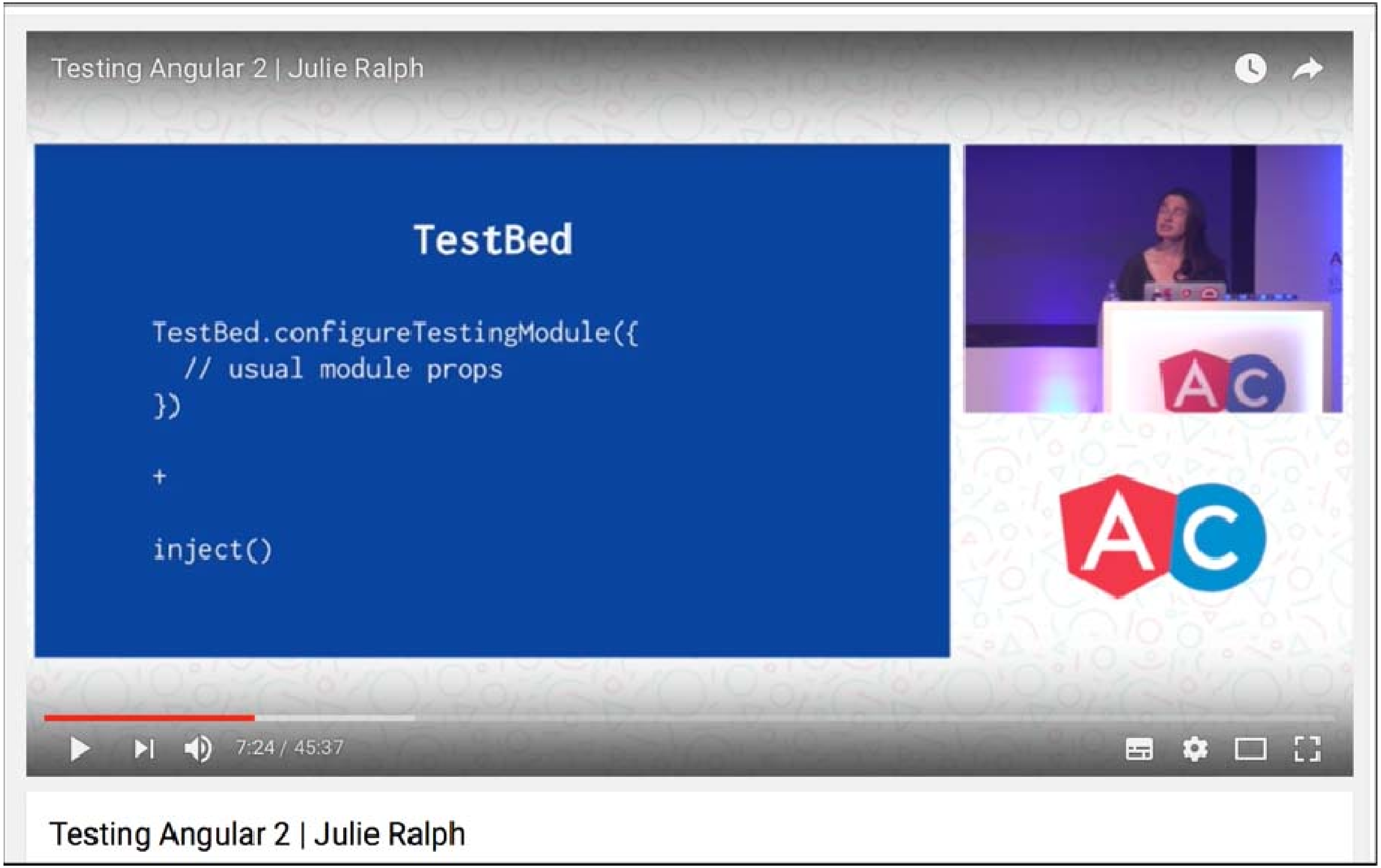
e2e	chore: base setup with CLI	a month ago
src	fix: missing parentheses	a month ago
.gitignore	chore: add .vscode to gitignore	a month ago
.travis.yml	chore: add travis config file	a month ago
README.md	docs: adjust intro	a month ago
angular-cli.json	chore: base setup with CLI	a month ago
karma.conf.js	feat: add karma mocha reporter	a month ago
package.json	chore(packages): upgrade TypeScript reference	a month ago
protractor.conf.js	chore: base setup with CLI	a month ago

https://github.com/juristr/angular-testing-recipes



```
describe('AppComponent', () => {
  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [
        AppComponent
      ],
    });
  });

  it('should create the app', async(() => {
    let fixture = TestBed.createComponent(AppComponent);
    let app = fixture.debugElement.componentInstance;
    expect(app).toBeTruthy();
  }));

  it(`should have as title 'app works!'`, async(() => {
    let fixture = TestBed.createComponent(AppComponent);
    let app = fixture.debugElement.componentInstance;
    expect(app.title).toEqual('app works!');
  }));
```

Videos on testing



Testing Angular 2 | Julie Ralph

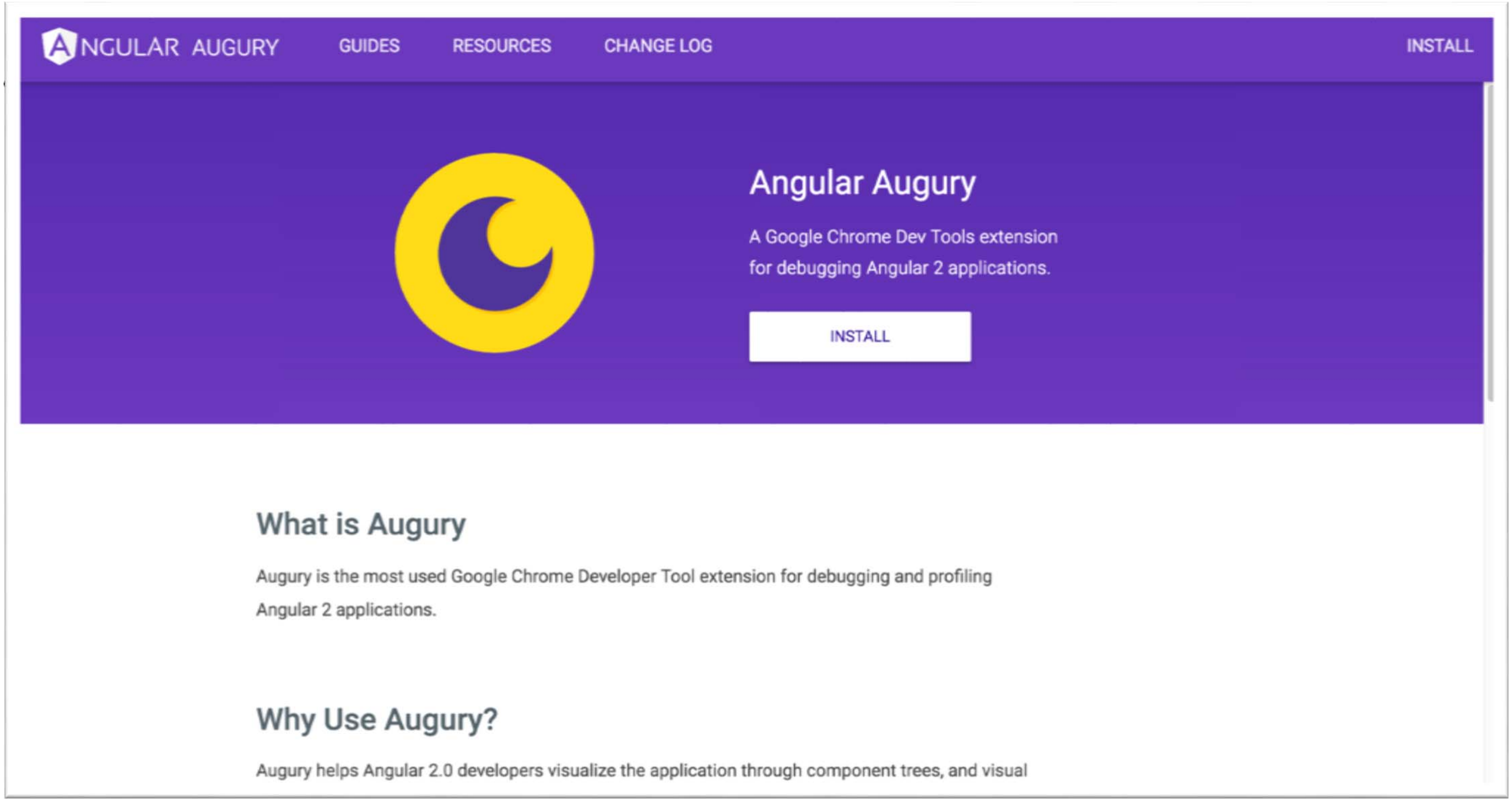
TestBed

```
TestBed.configureTestingModule({
  // usual module props
})

+

inject()
```

Testing Angular 2 | Julie Ralph

https://www.youtube.com/watch?v=f493Xf0F2yU `

Augury - debugging & analysis



Angular Augury

A Google Chrome Dev Tools extension for debugging Angular 2 applications.

What is Augury

Augury is the most used Google Chrome Developer Tool extension for debugging and profiling Angular 2 applications.

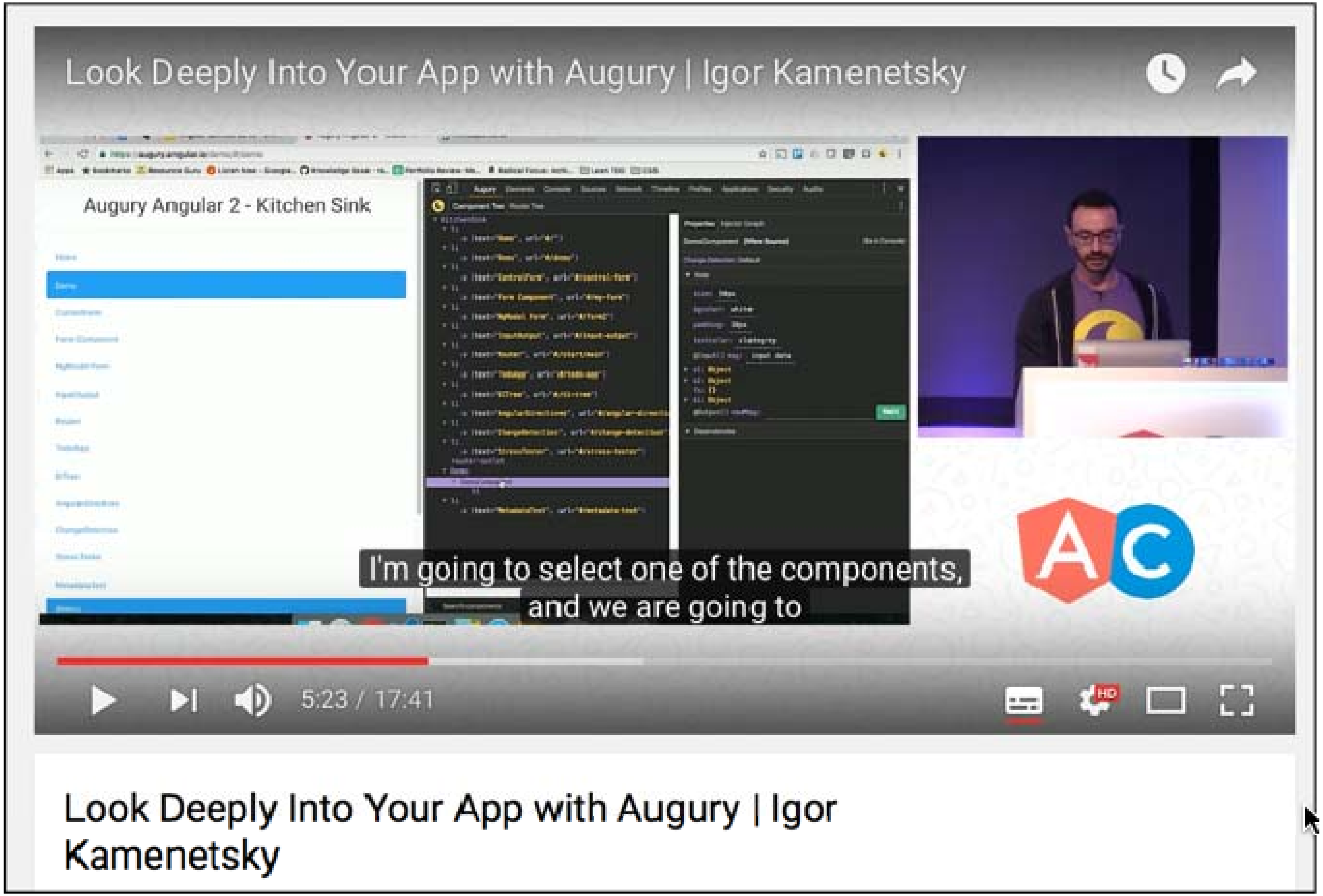
Why Use Augury?

Augury helps Angular 2.0 developers visualize the application through component trees, and visual

https://augury.angular.io/

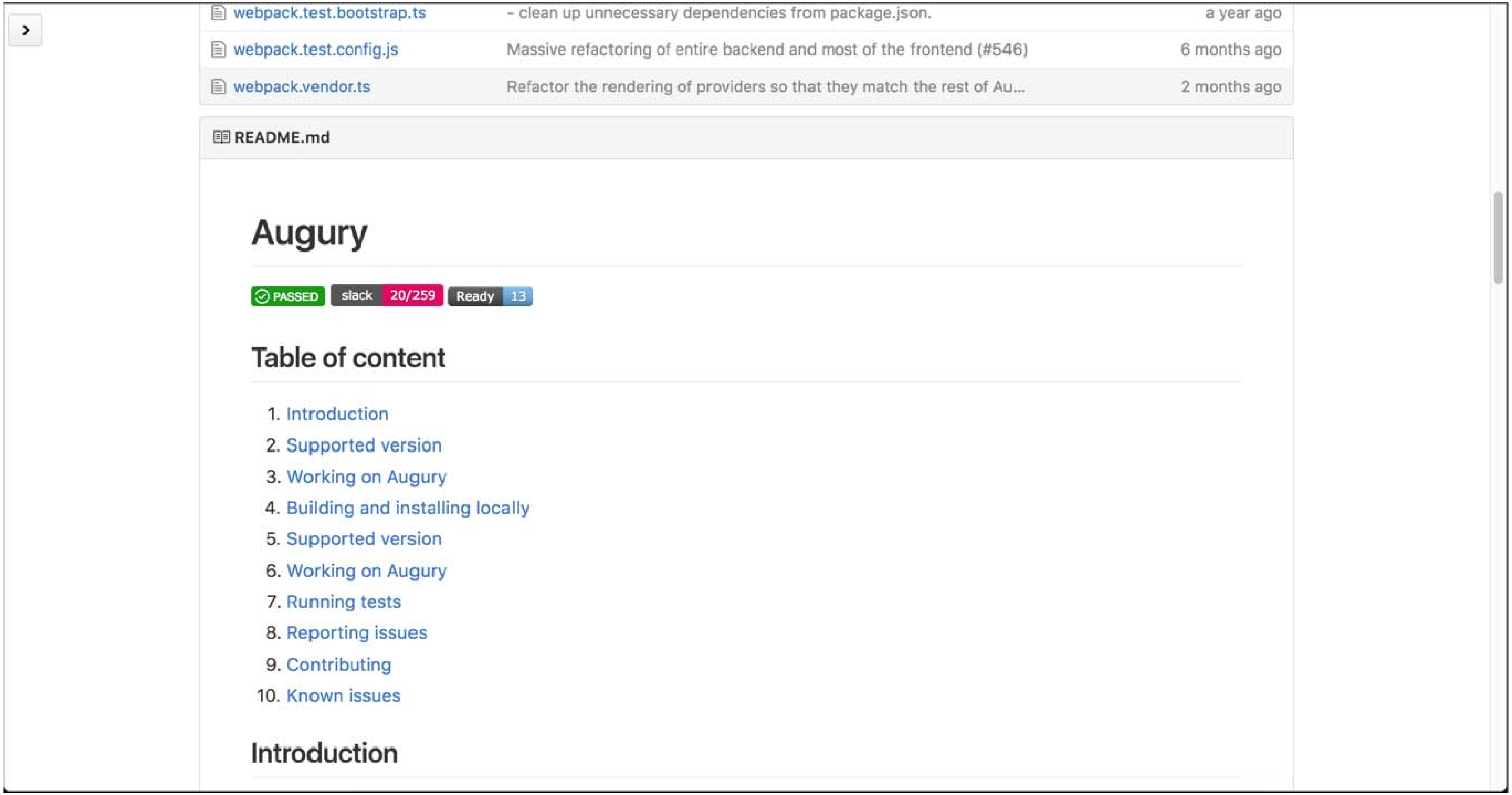
Extra

- Debugging – Augury
  - Extension to Chrome
- Insight in application architecture
- Insight in Routing structure
- Two themes – dark/light
- Insight in @ngModules / dependencies
- Install via Chrome Extions Store



Look Deeply Into Your App with Augury | Igor Kamenetsky
Augury Angular 2 - Kitchen Sink
I'm going to select one of the components,
and we are going to
5:23 / 17:41
HD
Look Deeply Into Your App with Augury | Igor Kamenetsky

https://www.youtube.com/watch?v=55vvJbrDgqw



webpack.test.bootstrap.ts	- clean up unnecessary dependencies from package.json.	a year ago
webpack.test.config.js	Massive refactoring of entire backend and most of the frontend (#546)	6 months ago
webpack.vendor.ts	Refactor the rendering of providers so that they match the rest of Au...	2 months ago

README.md

Augury

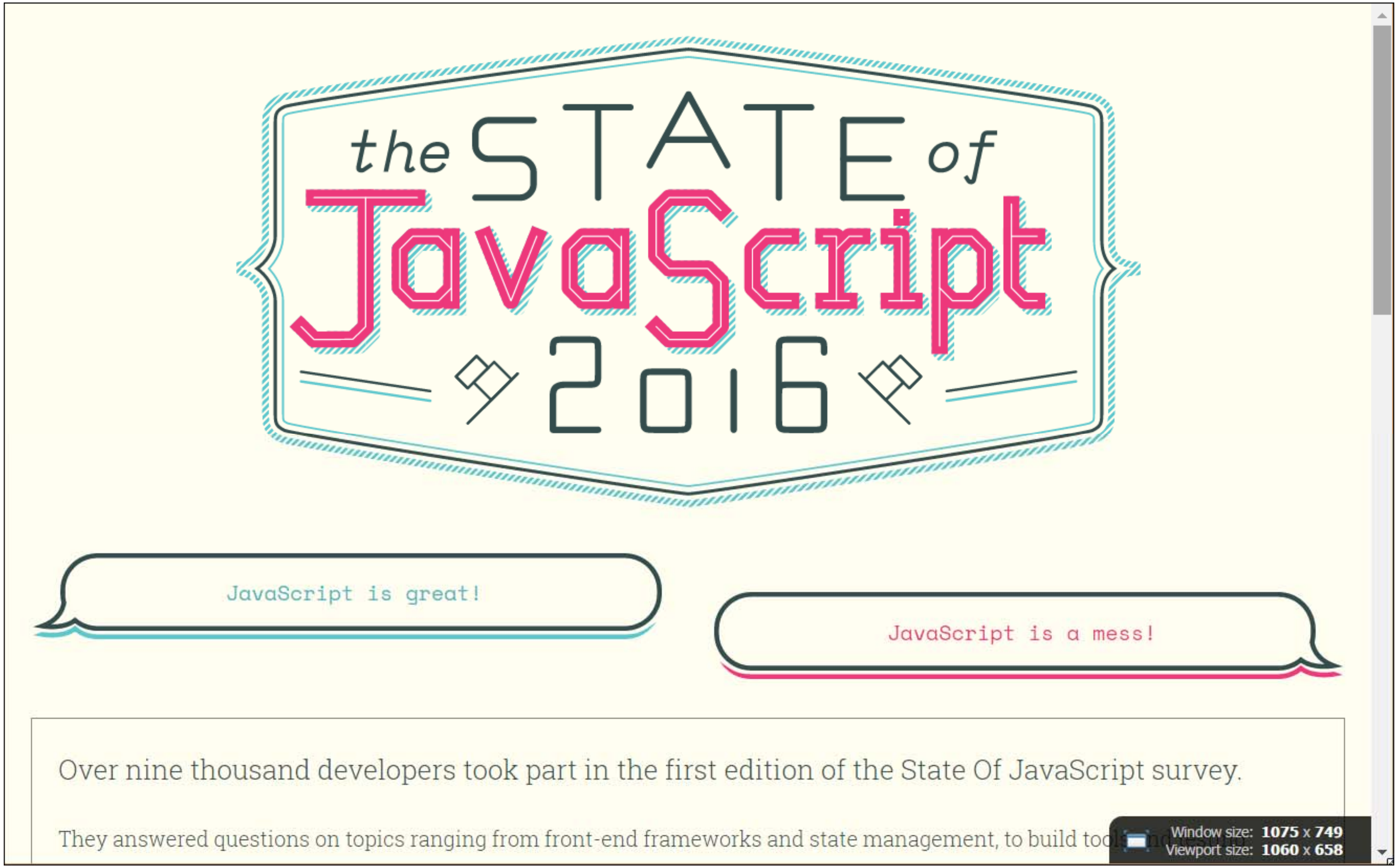
Table of content

1. Introduction
2. Supported version
3. Working on Augury
4. Building and installing locally
5. Supported version
6. Working on Augury
7. Running tests
8. Reporting issues
9. Contributing
10. Known issues

Introduction

https://github.com/rangle/augury

State of JavaScript



Over nine thousand developers took part in the first edition of the State Of JavaScript survey.

They answered questions on topics ranging from front-end frameworks and state management, to build tools

http://stateofjs.com/