Angular 2
Module 11 - Testing - WIP

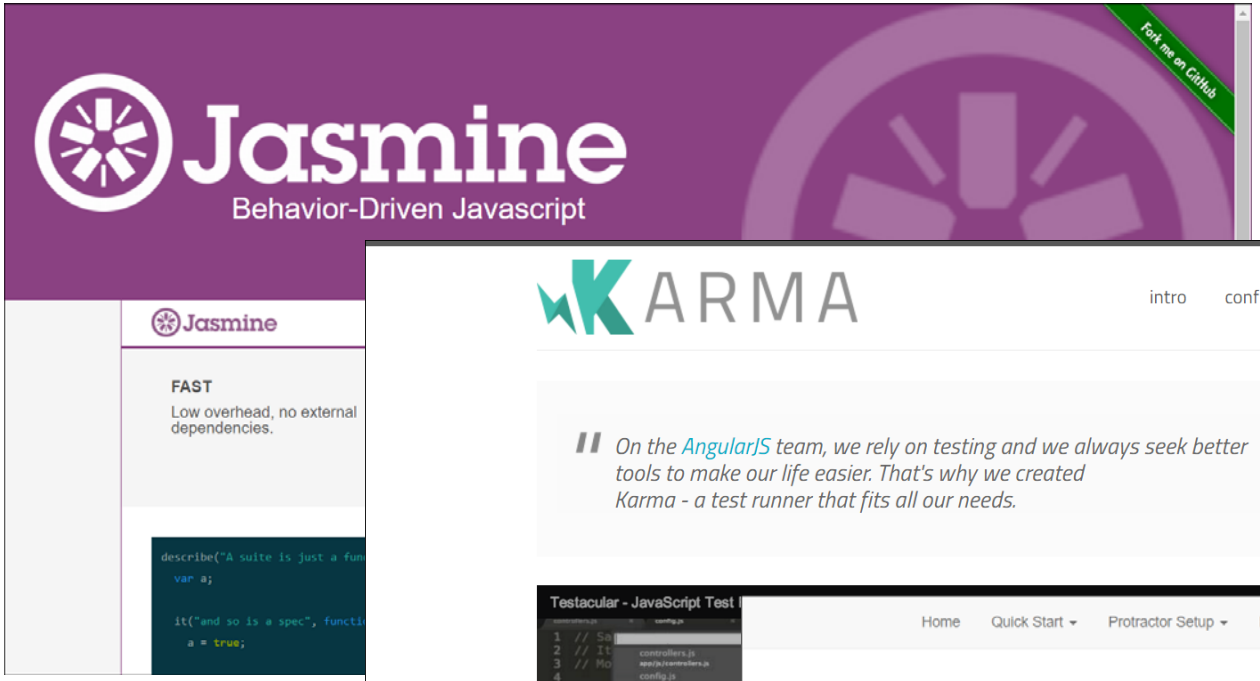
Peter Kassenaar –
info@kassenaar.com

Testing - some shortcuts

Goed introductie artikel

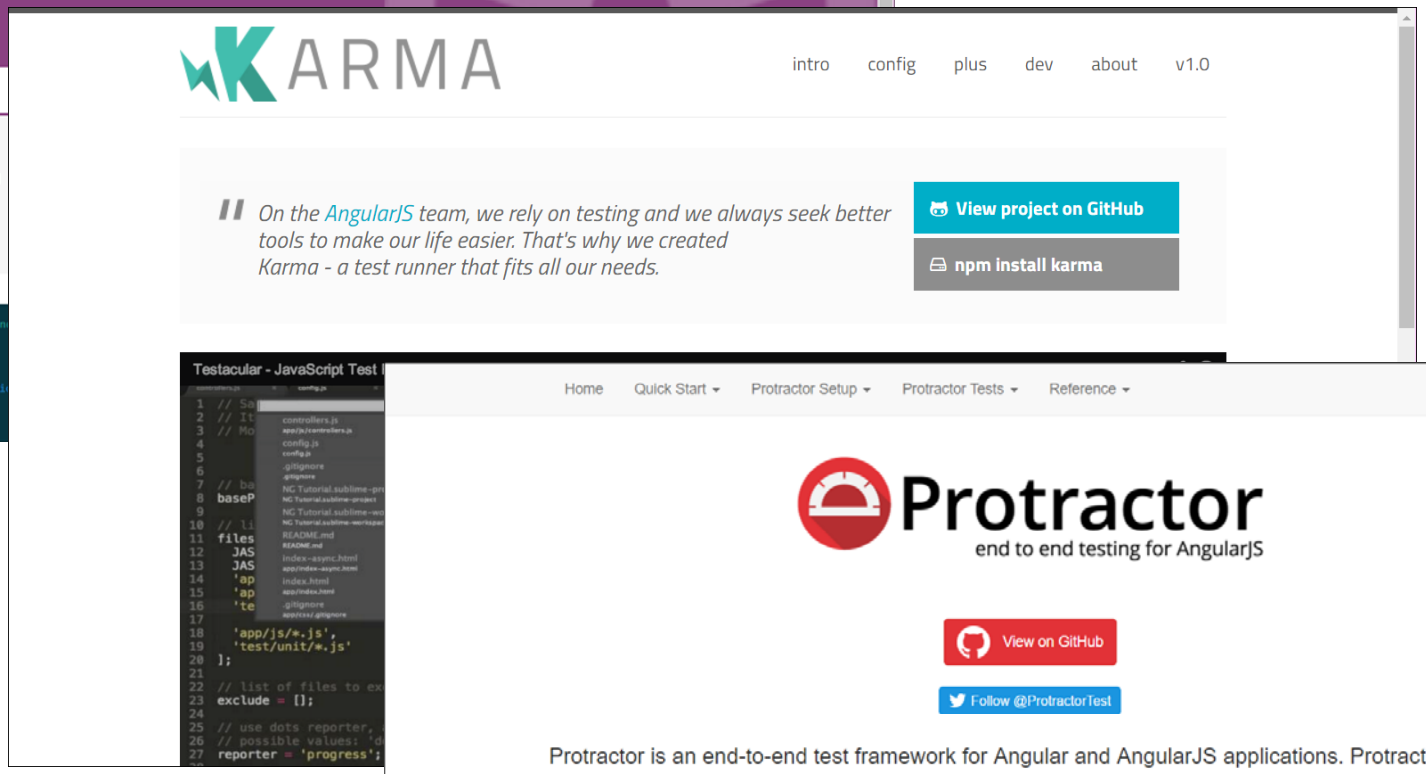
https://dzone.com/articles/talk-testing-with-angular-some-recipes

Tooling



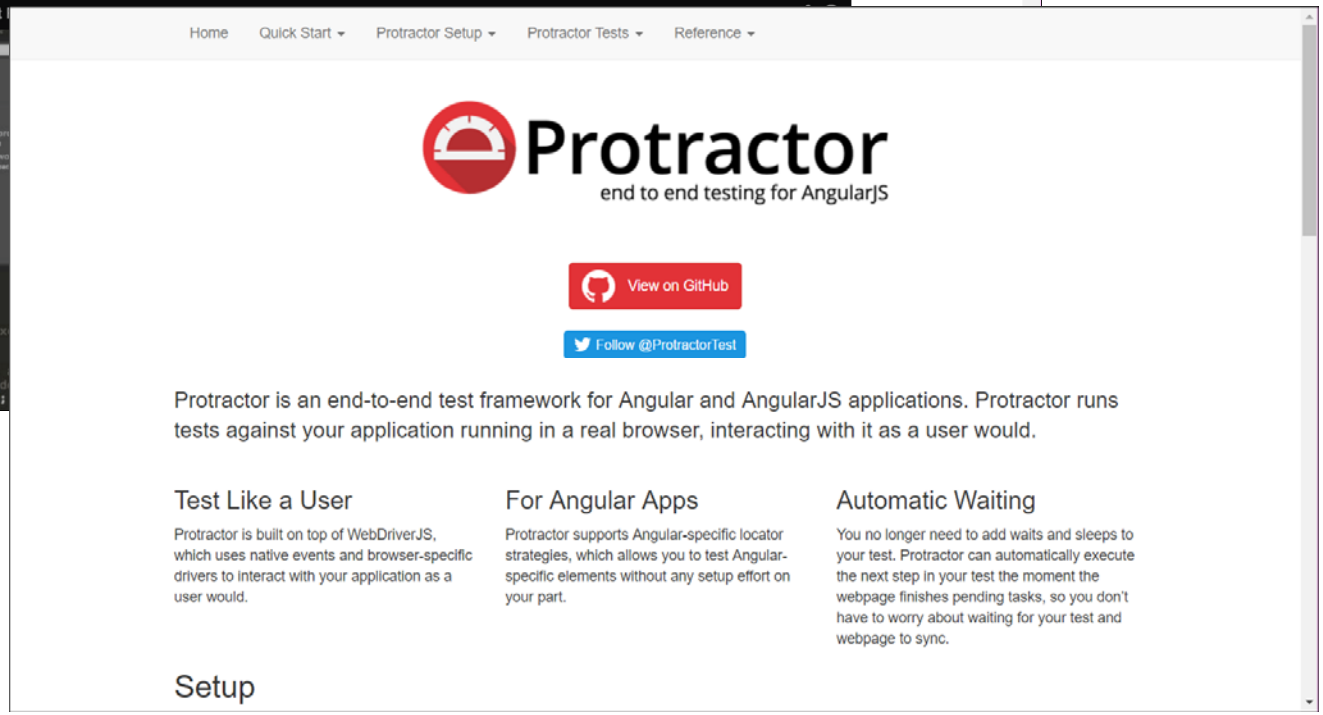
Jasmine:

Write Unit Tests & Specs



Karma:

Test Runner



Protractor:

End to end testing

- https://twitter.com/_victormejia/status/871971522033979392
- New, as of June 6, 2017 - Testing in Headless Chrome

https://jasmine.github.io/

https://karma-runner.github.io/1.0/index.html

http://www.protractortest.org/#/

General testing pattern / syntax

- **import** {Person} **from "./person.model"**;

- *// Person.model.ts*

```
export class Person {
  constructor(public name?: string,
              public email?: string) {
  }

  sayHello(): string {
    return `Hi, ${this.name}`;
  }
}
```

General unit test pattern

```
// Generic testing pattern

// 1. Describe block for every test suite
describe('The Person', () => {

   // 2. Variables used by this test suite
   let aPerson;

   // 3. Setup block, run before every individual test
   beforeEach(() => {
      aPerson = new Person('Peter');
   });

   // 4. Clean up after every individual test
   afterEach(() => {
      aPerson = null;
   });

   // 5. Perform each test in an it()-block
   it('should say Hello', () => {
      let msg = aPerson.sayHello();
      expect(msg).toBe('Hi, Peter');
   });

   // 6. More it()-blocks…
});
```

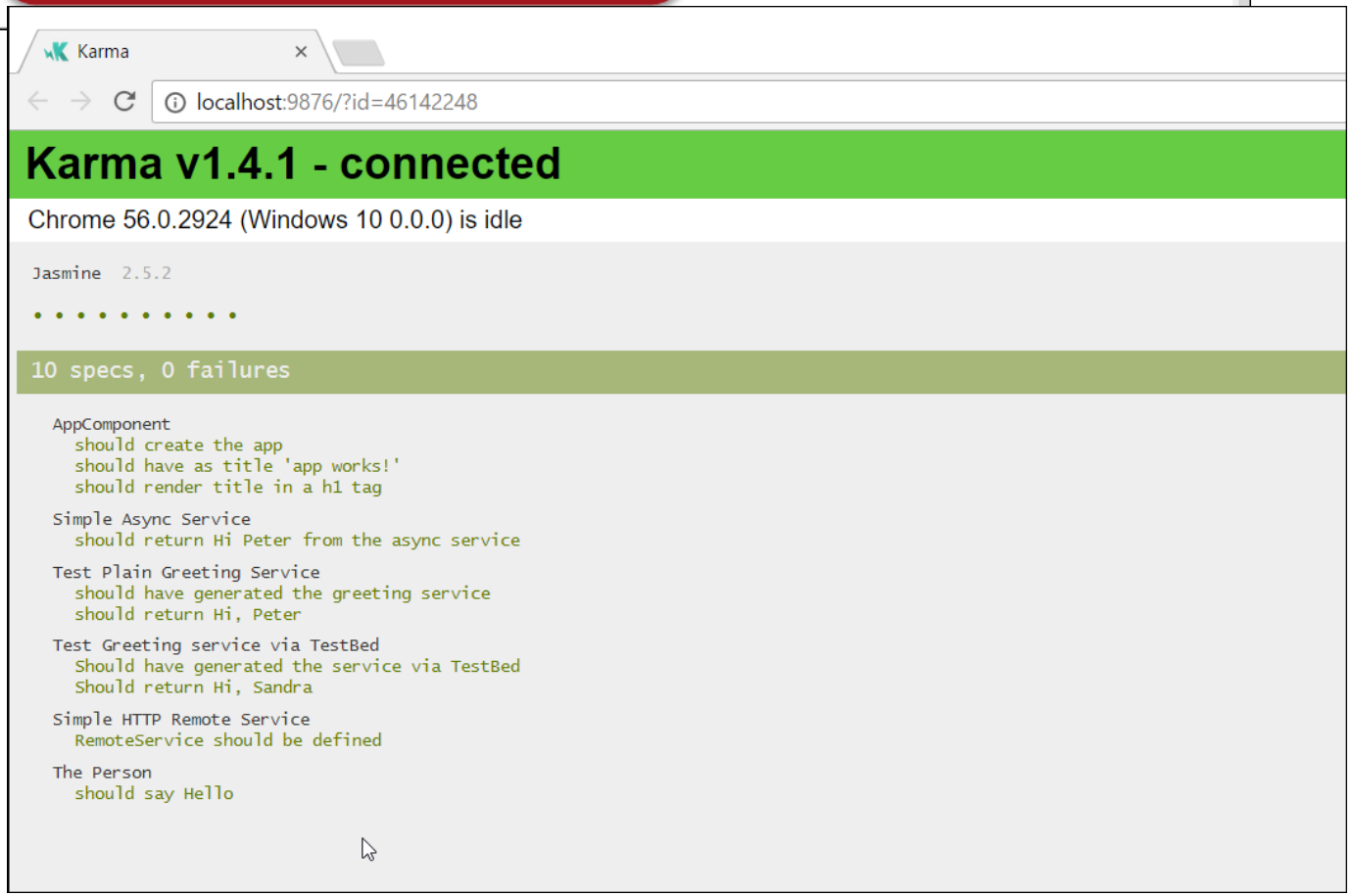
We're using angular-cli here

- Angular-cli: all dependencies already installed and configured.
  - Command: `ng test`
- Manual project? Install & setup karma and jasmine yourself
  - Create and adapt `karma.conf.js`
  - `karma --init`
  - Install and setup jasmine reporters
  - We're not doing that here.
  - See documentation

```
…
},
"devDependencies": {
   …
  "@types/jasmine": "2.5.38",
  "@types/node": "~6.0.60",
  "codelyzer": "~2.0.0",
  "jasmine-core": "~2.5.2",
  "jasmine-spec-reporter": "~3.2.0",
  "karma": "~1.4.1",
  "karma-chrome-launcher": "~2.0.0",
  "karma-cli": "~1.0.1",
  "karma-jasmine": "~1.1.0",
  "karma-jasmine-html-reporter": "^0.2.2",
  "karma-coverage-istanbul-reporter": "^0.2.0",
  "protractor": "~5.1.0",
   …
}
…
```

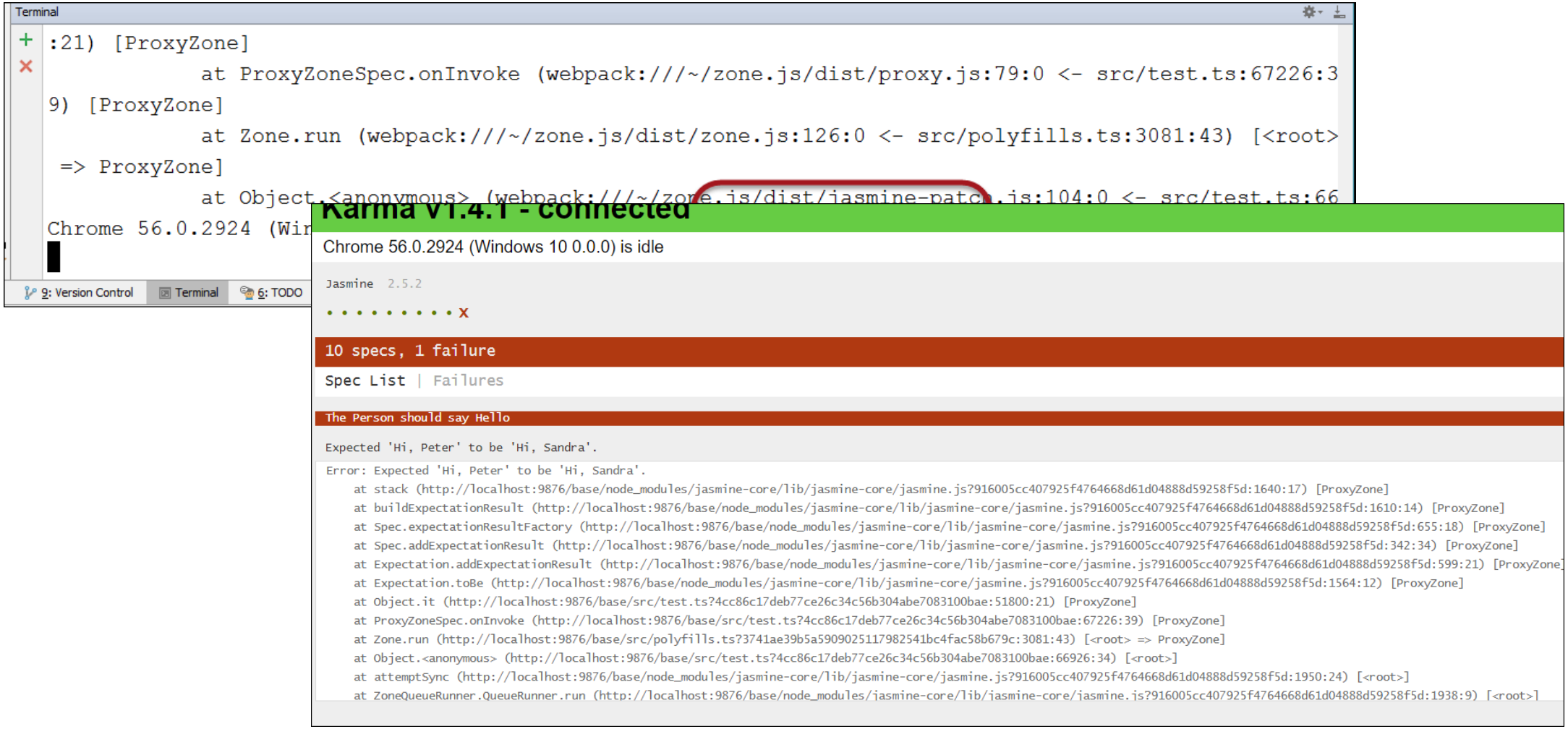
Browser like…

```
C:\Users\Peter Kassenaar\Desktop\ng-testing>ng test
10 03 2017 16:50:40.157:WARN [karma]: No captured browser, open http://localhost:9876/
10 03 2017 16:50:40.170:INFO [karma]: Karma v1.4.1 server started at http://0.0.0.0:9876/
10 03 2017 16:50:40.171:INFO [launcher]: Launching browser Chrome with unlimited concurrency
10 03 2017 16:50:40.310:INFO [launcher]: Starting browser Chrome
10 03 2017 16:50:42.220:INFO [Chrome 56.0.2924 (Windows 10 0.0.0)]: Connected on socket 3Ox3NIH4ohFYk
MAuAAAA with id 46142248
Chrome 56.0.2924 (Windows 10 0.0.0): Executed 10 of 10 SUCCESS (0.461 secs / 0.439 secs)
```



If anything goes wrong…

Hard to read output:



Solution: install custom reporters

Custom reporters

- Choices available.
- Editors choice: Karma Mocha Reporter
  - `npm install --save-dev karma-mocha-reporter`
- Edit karma.conf.js
  - `plugins: require('karma-mocha-reporter')`
  - `reporters: ? ['mocha', 'karma-remap-istanbul']`
    `:['mocha', 'kjhtml'],`
  - instead of progress

Run ng test again

Much nicer output and better error reports

```
START:
  AppComponent
    √ should create the app
    √ should have as title 'app works!'
    √ should render title in a h1 tag
  Simple Async Service
    √ should return Hi Peter from the async service
  Test Plain Greeting Service
    √ should have generated the greeting service
    √ should return Hi, Peter
  Test Greeting service via TestBed
    √ Should have generated the service via TestBed
    √ Should return Hi, Sandra
  Simple HTTP Remote Service
    √ RemoteService should be defined
  The Person
    √ should say Hello

Finished in 0.485 secs / 0.46 secs @ 17:49:52 GMT+0100 (West-Europa (standaardtijd))

SUMMARY:
√ 10 tests completed
```

...or, when errors occur

```
SUMMARY:
√ 9 tests completed
× 1 test failed

FAILED TESTS:
  The Person
    × should say Hello
      Chrome 56.0.2924 (Windows 10 0.0.0)
    Expected 'Hi, Peter' to be 'Hi, Sandra'.
        at Object.it (webpack:///src/app/tests/generic-pattern.spec.ts:23:14 <- src/test.ts:51800:21
 [ProxyZone]
```

Terms and conditions...

- TestBed
- inject
- async
- fakeAsync
- ComponentFixture
- DebugElement
- configureTestingModule

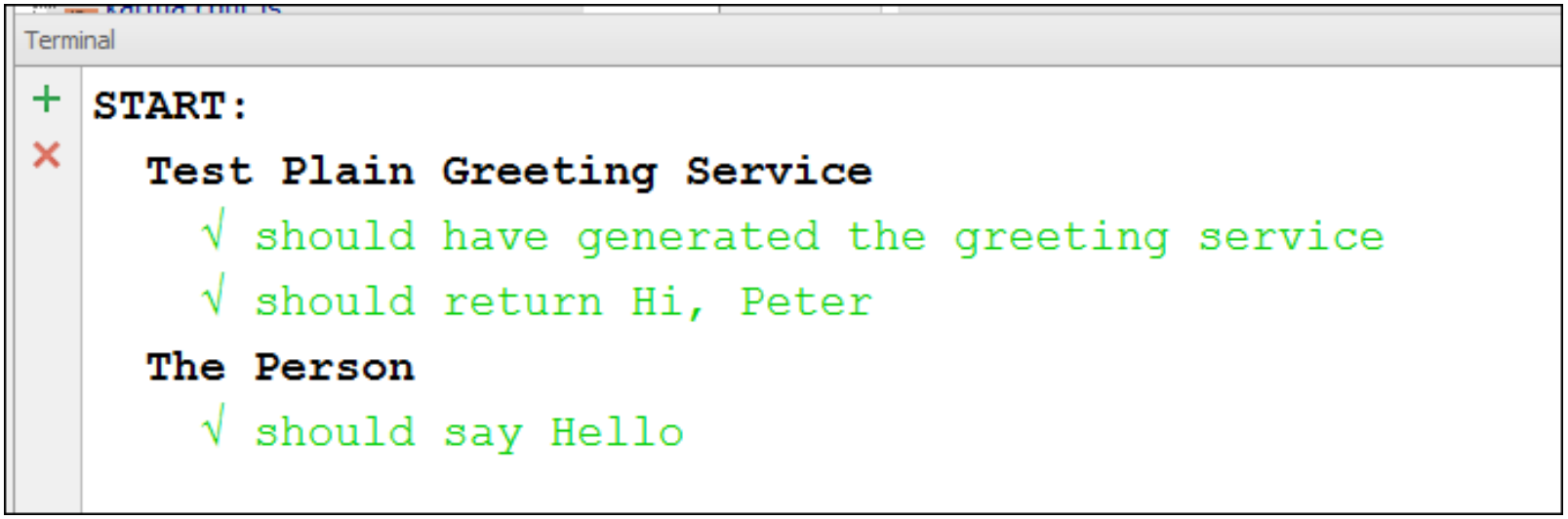
Testing a single service

One of the more easier concepts to test. So let's start there.

```
// greeting.service.ts
import {Injectable} from '@angular/core';

@Injectable()
export class GreetingService {

   constructor() {
   }

   greet(name: string): string {
      return `Hi, ${name}`;
   }
}
```

```
// greeting.service.spec.ts
import {GreetingService}from './greeting.service';
describe('Test Plain Greeting Service', () => {
    let greetingService;
    beforeEach(() => {
        greetingService = new GreetingService();
    });

    it('should have generated the greeting service', () => {
        expect(greetingService).toBeTruthy()
    });

    it('should return Hi, Peter', () => {
        let msg = greetingService.greet('Peter');
        expect(msg).toEqual('Hi, Peter');
    });
});
```

Output


Terminal
START:
Test Plain Greeting Service
√ should have generated the greeting service
√ should return Hi, Peter
The Person
√ should say Hello

But what about DI?

// HIER VERDER

But what about DI?

- Most of the time you don’t test a single service.
- Angular relies on DI for your services and uses them in the context of an @ngModule – show simple Module.
- In testing this is referred to as the TestBed.
- A TestBed takes a similar configuration object to mimic a module
- TestBed.configureTestingModule({…., providers:[<services>]})
- In TestBed you don’t need to configure the complete module. Only the parts you need. In this case the greetingService.
- Then instantiate the service from the TestBed. All possible other dependencies are resoveld. service = TestBed.get(GreetingService);
- Alternative notation: via injector, instead of TestBed.get().
- beforeEach(inject([GreetingService], (s:GreetingService)=>{ service = s}));

Async behavior

- If the services uses async calls, for instance promises or Observables
- Create service with Async call, for instance new Promise which resolves after 100ms.
- In spec-file, it always returns true, because the expect-statement is not run in the .then()-clause
- service.greetAsync(**'Peter'**).then((result)=>{
  expect(result).toEqual(**'Hi, Peter12234'**);
  }) // test passed
- Solution: wrap it inside Angular async-construct, like
- it(**'should return Hi Peter from the async service'**, async(()=>{
  service.greetAsync(**'Peter'**).then((result)=>{
  expect(result).toEqual(**'Hi, Peter12234'**);
  })
  })) // Test failed.
- Also available : fakeAsync. More control. Less often used

Testing XHR calls

- Setup remote service to fetch some data over http
- *// Get fake people*

```
getPeople():Observable<any>{
    return
this.http.get('http://someEndPoint/somePeople.json')
        .map(result =>result.json());
}
```

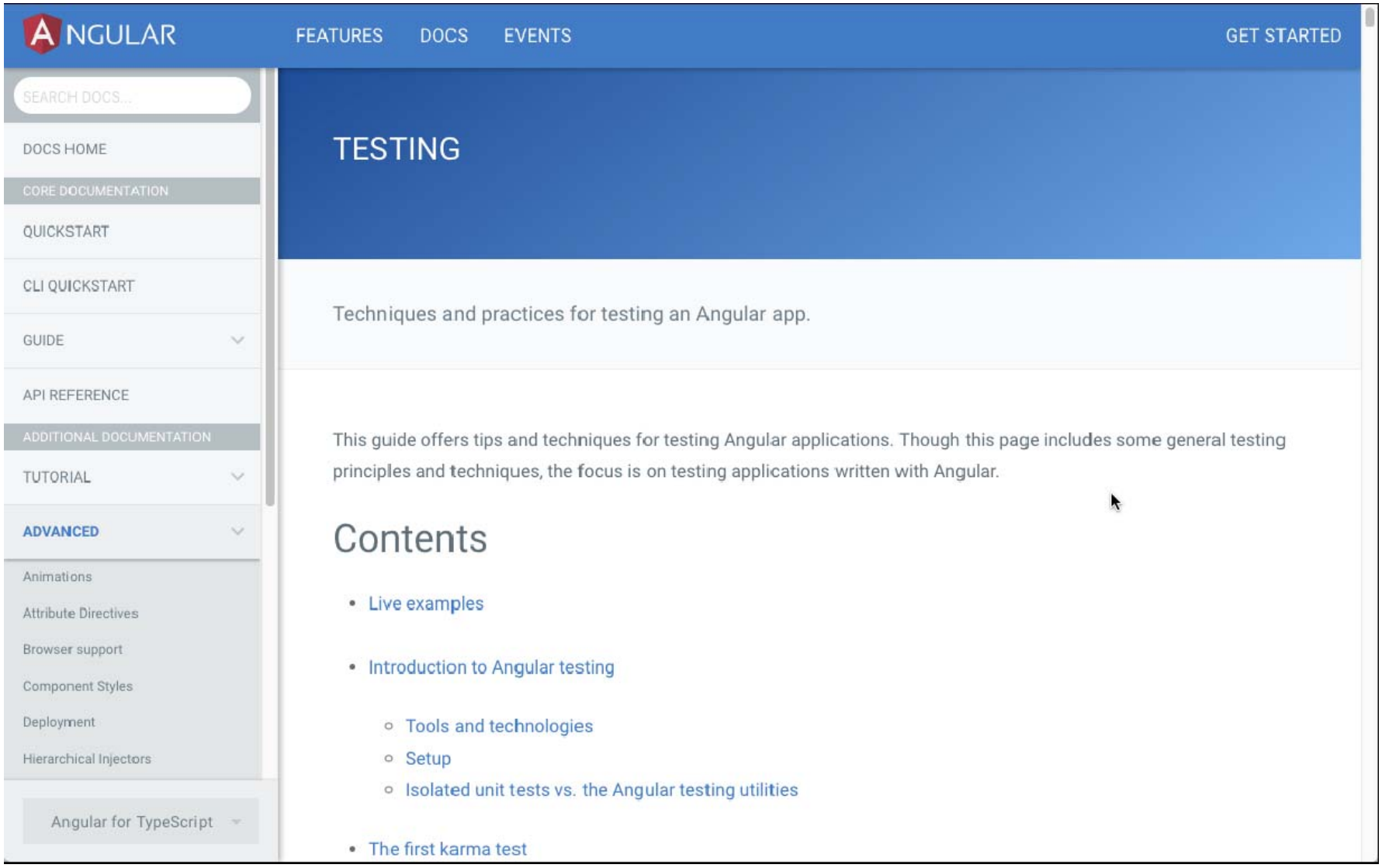
It will fail

- Configure the test like so:
- beforeEach(()=>{
  TestBed.configureTestingModule({
  imports:[HttpModule],
  providers:[RemoteService]
  })
  service = TestBed.get(RemoteService);
  });
- it('**Should return people.json**',async(()=>{
  let people;
  service.getPeople().subscribe((result)=>{
  people = result;
  });
  expect(people).toBeDefined();
  }))

Solution: create Mock Backend

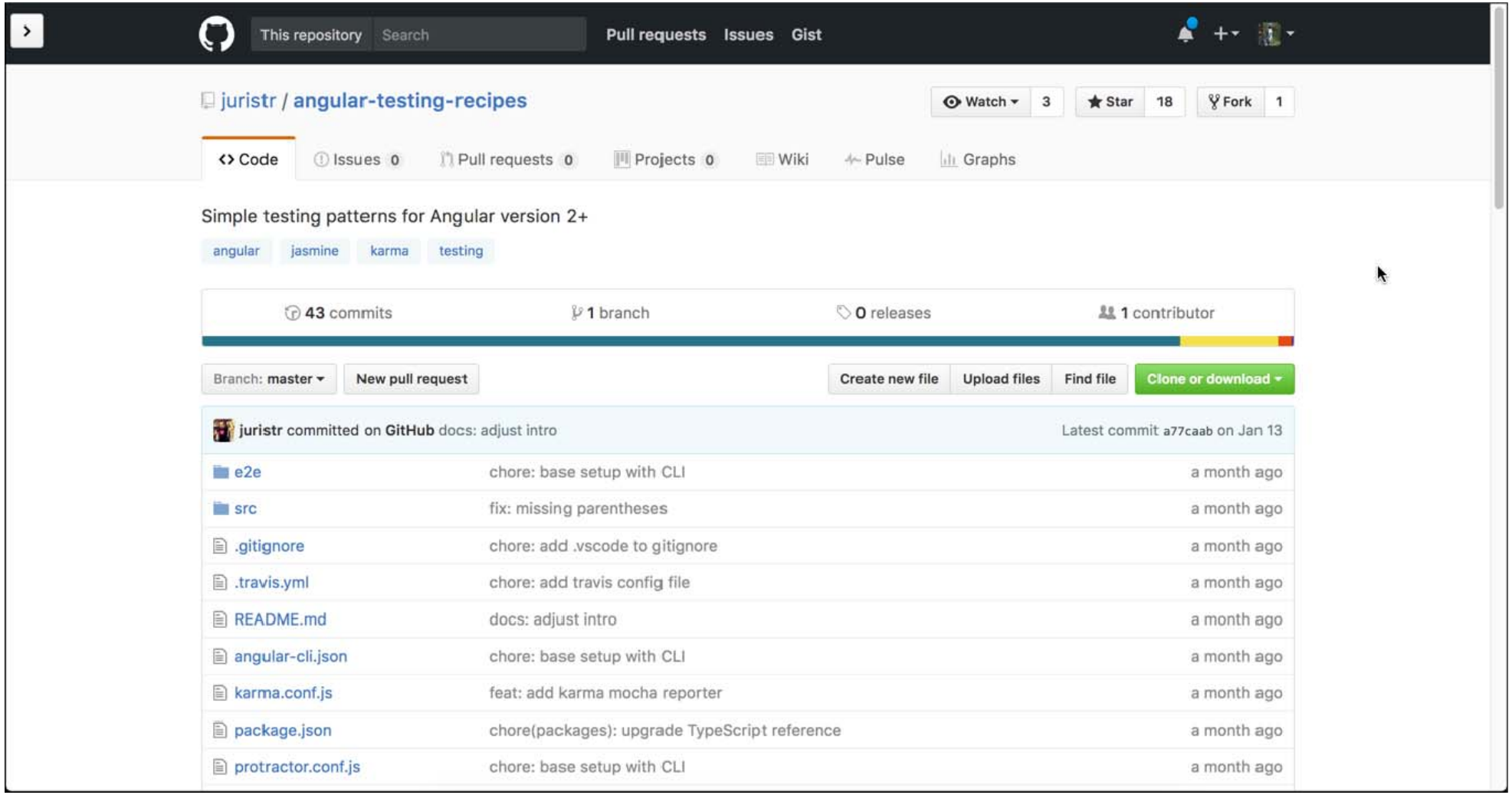
- Hier verder….

Testing documentation



https://angular.io/guide/testing

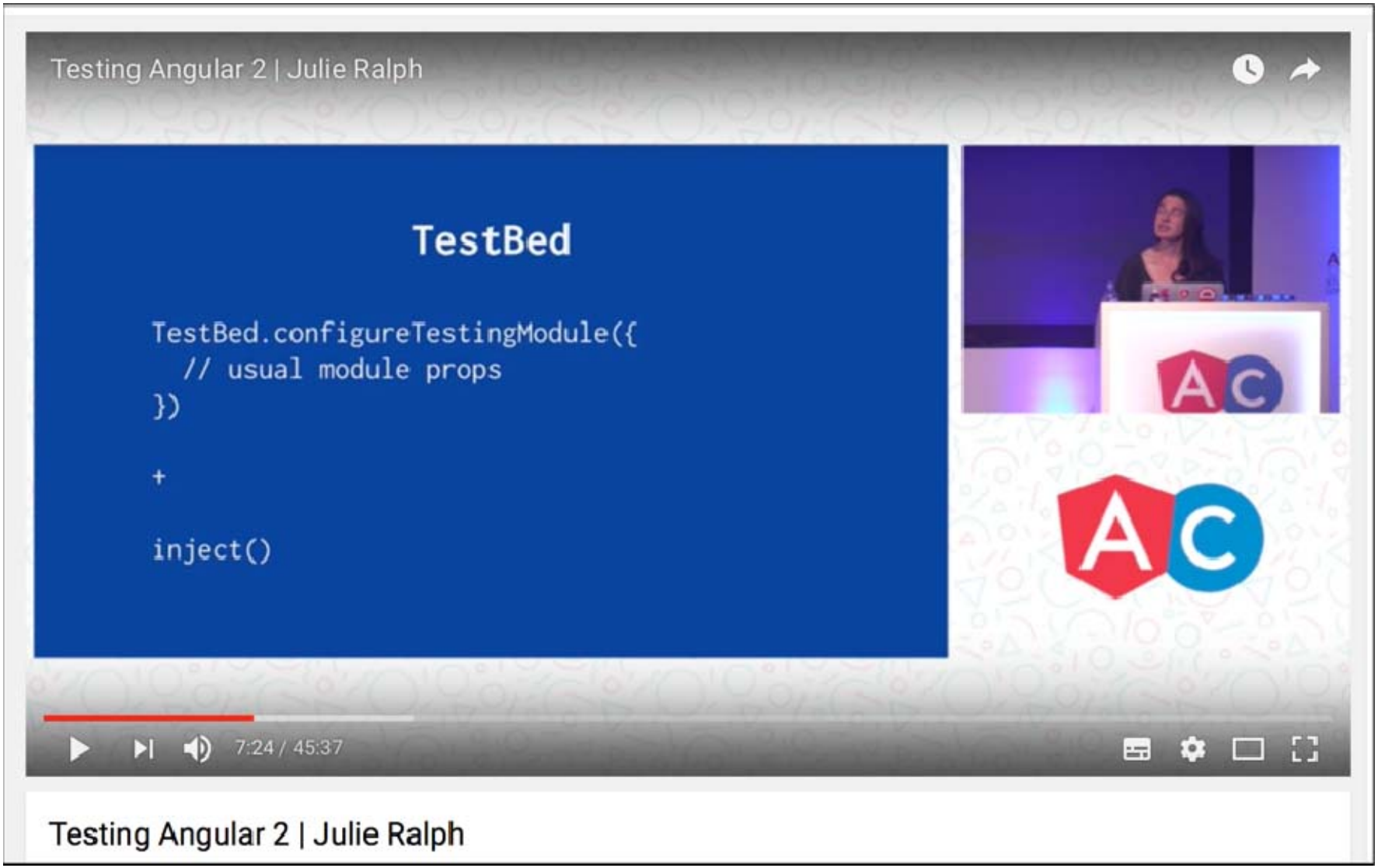
Repo – Angular Testing recipes



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

```
6   describe('AppComponent', () => {
7     beforeEach(() => {
8       TestBed.configureTestingModule({
9         declarations: [
10          AppComponent
11        ],
12      });
13    });
14
15    it('should create the app', async(() => {
16      let fixture = TestBed.createComponent(AppComponent);
17      let app = fixture.debugElement.componentInstance;
18      expect(app).toBeTruthy();
19    }));
20
21    it(`should have as title 'app works!'`, async(() => {
22      let fixture = TestBed.createComponent(AppComponent);
23      let app = fixture.debugElement.componentInstance;
24      expect(app.title).toEqual('app works!');
25    }));
```

Videos on testing



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

Goed introductie artikel

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