Test All The Things

Getting Started with Angular Unit and E2E Tests

Tyler S. Lemke
linkedin.com/in/tylerslemke
https://www.youtube.com/c/TylerSLemke

Check out ZenLeap (startup I work for)

- Simplified Hiring
- No Resumes
- Inexpensive

http://bit.ly/get-zenleap

Get this PDF

http://bit.ly/start-angular-testing

What's up with tonight?

- Why automated tests are awesome
- Unit Testing
- End to End Tests
- Alternative Tools and Frameworks
- Best Practices

Why Automate Tests?

- Manual Testing Sucks
- Unit Tests Provide a Contract/Documentation
- Check Regressions
- E2E Tests Tell you if your app works

Why aren't you testing now?

- Testing is a skill
- You might feel overwhelmed
- You don't know where to start

How can we unit test in Angular?

Using Jasmine Framework

Karma Test Runner

Let's Start Unit Testing

1. Please download the following
   a. https://github.com/tylerslemke/Angular-FrankenTest
2. Open in VSCode
3. Run npm install

Jasmine

> Jasmine is a behavior-driven development framework for testing JavaScript code. It does not depend on any other JavaScript frameworks. It does not require a DOM. And it has a clean, obvious syntax so that you can easily write tests.

Jasmine Code Example

```
describe("A suite is just a function", function() {
  var a;

  it("and so is a spec", function() {
    a = true;

    expect(a).toBe(true);
  });
});`
```

Angular TestBed

> The TestBed is the most important of the Angular testing utilities. The TestBed creates a dynamically-constructed Angular test module that emulates an Angular @NgModule.

Lets create a test from scratch

go to src/app.yourfirstspec.ts

Jest (Alternative to Jasmine)

- Faster
- More Stable
- Snapshot Testing

How can we e2e test in Angular?

Protractor

Build on WebDriverJS

Lets do a "fake" e2e test in Protractor

Nevermind. Lets not.

https://www.protractortest.org/#/tutorial

Cypress Instead

Story 1

> 'It (Cypress) is better in every way... I spent 2 days trying to write tests for a project and they were not working, I converted them and had them working in cypress in 2 hrs (Paraprhase)' - Alain Chautard (Google Developer Expert)

Cypress Instead

Story 2

> '[some] People will write huge libraries of protractor tests and ultimately end up abandoning them' - Jesse Sanders

Cypress Instead

Story 3

> At a high level, the primary reason for this switch was because over the course of 2 months, our developers wrote more e2e tests and got more value out of Cypress than the previous 2 years combined with Protractor."
> ~ Jeff Whelpley, CTO at Swish, Google Developer Expert, co-creator of Angular Universal

Caveat Emptor

Cypress is great but consider Protractor:

- Multi-browser support
- Other situations

Let's Demo Cypress

1. In one terminal run the "npm start" command
2. In another terminal, run "npx cypress open"
3. Also, open cypress/integration/test.spec.js

Did you notice anything about the selectors in our tests?

```
cy.get('h1').should("contain", "Tour of Heroes");
```

```
cy.get('.search-result').should("contain", "Bombasto");
```

Best Practice

Decouple your test selectors

```
cy.get('h1').should("contain", "Tour of Heroes");
```

```
cy.get('[data-test]=""').should("contain", "Bombasto");
```

How can you turn this code?

```
import { TestBed, async, ComponentFixture } from '@angular/core/testing';
import { ButtonComponent } from './button.component';
import { Component, DebugElement } from "@angular/core";
import { By } from "@angular/platform-browser";

describe('ButtonComponent', () => {
  let fixture: ComponentFixture<ButtonComponent>;
  let instance: ButtonComponent;
  let debugElement: DebugElement;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ ButtonComponent ]
    })
    .compileComponents();

    fixture = TestBed.createComponent(ButtonComponent);
    instance = fixture.componentInstance;
    debugElement = fixture.debugElement;
  }));

  it('should set the class name according to the [className] input', () => {
    instance.className = 'danger';
    fixture.detectChanges();
    const button = debugElement.query(By.css('button')).nativeElement as HTMLButtonElement;
    expect(button.classList.contains('danger')).toBeTruthy();
    expect(button.classList.contains('success')).toBeFalsy();
  });

});
```

Into this?

```
import { ButtonComponent } from './button.component';
import { Spectator, createTestComponentFactory } from '@netbasal/spectator';

describe('ButtonComponent', () => {

  let spectator: Spectator<ButtonComponent>;
  const createComponent = createTestComponentFactory(ButtonComponent);

  beforeEach(() => spectator = createComponent());

  it('should set the class name according to the [className] input', () => {
    spectator.setInput('className', 'danger');
    expect(spectator.query('button')).toHaveClass('danger');
    expect(spectator.query('button')).not.toHaveClass('success');
  });
});
```

Spectator

Written on top of Angular Testing Framework

- Less Boilerplate
- Cleaner API
- Schematics
- Custom Matchers
- Jest/Jasmine Support

https://github.com/NetanelBasal/spectator

TDD with Wallaby.js

- Only runs one test at a time
- Can be finicky
- No other tool out there like it
- Get Wallaby (Referral link supports me making these presentations)

General Resources

- https://angular.io/guide/testing
- Pluralsight - Unit Testing in Angular
- Pluralsight - Play by Play: Fundamentals of Angular Testing
- https://blog.angulartraining.com/how-to-write-unit-tests-for-angular-code-that-uses-the-httpclient-429fa782eb15

Jest Resources

- https://itnext.io/how-to-use-jest-in-angular-aka-make-unit-testing-great-again-e4be2d2e92d1
- https://jestjs.io/docs/en/snapshot-testing

Cypress Resources

- https://www.cypress.io/blog/2018/03/20/angular-cypress-love/
- https://www.youtube.com/watch?v=GH9Dvo_BYkk
- https://cypress-io.ghost.io/blog/2019/01/03/stop-using-page-objects-and-start-using-app-actions/
- https://medium.com/reactbrasil/deep-diving-pageobject-pattern-and-using-it-with-cypress-e60b9d7d0d91

Spectator Resources

- https://engineering.datorama.com/spectator-for-angular-or-how-i-learned-to-stop-worrying-and-love-the-spec-2aa8521c8488
- https://github.com/NetanelBasal/spectator

Wallaby Resources

- Get Wallaby https://wallabyjs.com/?referrer=tylerslemke (Referral link supports me making these presentations)
- Install Wallaby
- Pluralsight - Play by Play: Fundamentals of Angular Testing