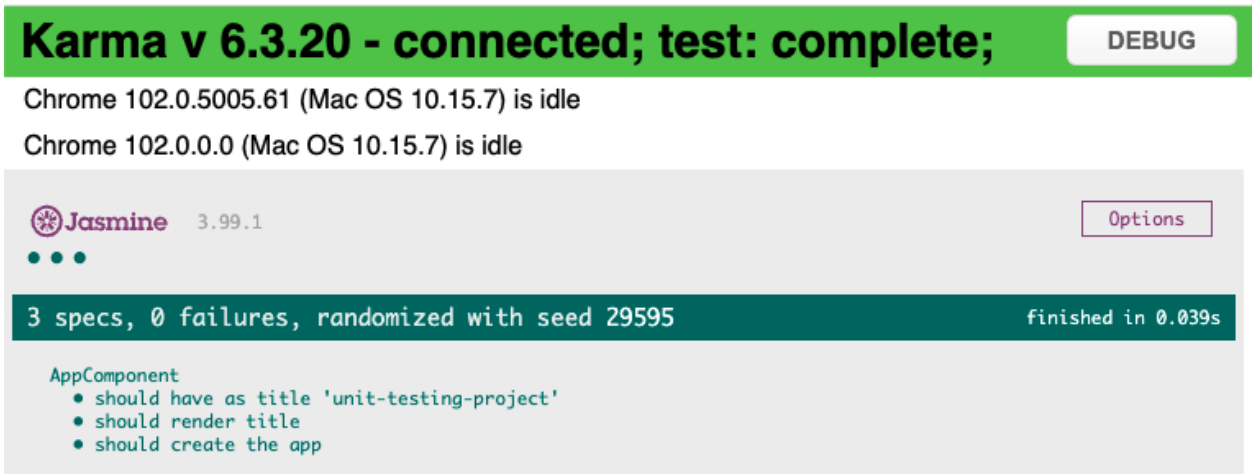
#angular: Unit test an Angular App

3 Reasons why unit testing is required?

- Guard against breaking changes
- Analyze code behavior
- Reveal Design mistakes



Yes, the attached image shows the GUI when we run unit tests for an angular cli project.

It's very simple, create angular app using

- ng new my-app

Now run

- ng test

And you see the GUI when you open the url given in the console.

Before diving into code, let's talk about tools that are used by angular to help us write unit tests:

1. Jasmine -> It is the testing framework
2. Karma -> It is a test runner for running our unit tests.
3. Angular testing utilities -> A set of helper methods that assist us in setting up our unit tests.

By Default when you create an angular app, it comes with 3 units already written for an AppComponent. Open app.component.spec.ts file and you will see following code:

```
import { TestBed } from '@angular/core/testing';
```

// TestBed class -> Configures and initializes the environment for unit testing and provides methods for creating components and services in unit tests.

```
import { AppComponent } from './app.component';

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

// beforeEach Block is executed before the execution of each test case.
// Now What is the test case?
// Test case is it Block.
// it methods takes 2 arguments, a string that describes your gtest case and a callback function, which is actually a unit test function

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

// What is happening in the above “it” method?
// Above test case make sure the appComponent is created successfully.

```
 it(`should have as title 'unit-testing-project'`, () => {
   const fixture = TestBed.createComponent(AppComponent);
   const app = fixture.componentInstance;
   expect(app.title).toEqual('unit-testing-project');
 });
```

// What is happening in the above “it” method?
// Above test case make sure the appComponent has property title and that property is equal to string `'unit-testing-project'`

```
 it('should render title', () => {
```

```
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
      expect(compiled.querySelector('.title')?.textContent).toCon
tain('unit-testing-project app is running!');
 });
```

// What is happening in the above “it” method?
// Above test case make sure the appComponent template file has a n html element, with class name equal to .title and the element contains the text equal to `'unit-testing-project app is running!'`

```
});
```

I hope it makes little sense and gives a good start to explore further about unit testing of your angular app.

For further reading, follow below links:
https://angular.io/guide/testing
https://semaphoreci.com/community/tutorials/testing-components-in-angular-2-with-jasmine

Please add your feedback in the comment section.

follow and add Rizwan Mushtaq Dhudhaal 👨‍💻 in your friend list