Unit Test in Angular2 :

Each block starting with it

1)Why we need unit test in angular2 ?

Ans : This to to test the following scenerios :

a)Does the component is working as intended

b)Does the pipe is working as intended

c)Does the service works as intended

d)Does input works as intended

e)Does injection works as intended

2)What are describe and it in angular 2 spec file ?

Ans : All the it block are test in itself.All it blocks are independent of each other.

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

//TestBed is testing utility object.

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

describe('Something..',()=>{

beforeEach(()=>{

TestBed.configureTestingModule({ //This allows us to configure module for our testing.It tells which component we have in this testing environment

declarations : [

AppComponent

]

});

});

it('should create the app',async(()=>{

let fixture = TestBed.createComponent(AppComponent)

let app = fixture.debugElement.componentInstance; // debugElement is an element exposed to us for the testing purposes.

expectA(app).toBeTruthy() // It should exist

})

it('should have title app works',async(()=>{

let fixture = TestBed.createComponet(AppComponent);

let app = fixture.debugElement.componentInstance;

expect(app.title).toEqual('app works')

})

);

it('should render title in h1 tag',async(()=>{

let fixture = TestBed.createComponent(AppComponent);

fixture.detectChanges();

let compiled = fixture.debugElement.nativeElement;

expect(compiled.querySelector('h1').textContent).toContain('app works');

})

);

it('')

Running test though cli :

Command is ng-test

Create a new Component with name user. It should contain following files

a)user.component.ts

b)user.html

c)user.css

d)user.spec.ts

Testing dependencies. Component and services.

How to get the reference of service inside the spec file ?

Ans :

var app = fixture.debugElement.componentInstance;

var userService = fixture.debugElement.injector.get(UserService);

fixture.detectChanges();

expect(userService.user.name).toEqual(app.user.name)

Testing http asynchronous request in angular 2 testing :

**data.service.ts**

export class DataService{

getData(){

var dataProvise = new Promise((resolve,reject)=>{

setTimeout(()=>{

resolve('data');

},2000);

})

return dataProvise;

}

}

**Spec file :**

/\* tslint:disable:no-unused-variable \*/

import { async, ComponentFixture, TestBed,tick } from '@angular/core/testing';

import { By } from '@angular/platform-browser';

import { DebugElement } from '@angular/core';

import {UserService} from './user.service';

import { UserComponent } from './user.component';

import {CalculatorService} from './calculator.service';

import {DataService} from './data.service';

describe('UserComponent', () => {

let component: UserComponent;

let fixture: ComponentFixture<UserComponent>;

beforeEach(async(() => {

TestBed.configureTestingModule({

declarations: [ UserComponent ]

})

.compileComponents();

}));

beforeEach(() => {

fixture = TestBed.createComponent(UserComponent);

component = fixture.componentInstance;

fixture.detectChanges();

});

/\*it('should create', () => {

expect(component).toBeTruthy();

});\*/

/\*it('Service name should be equal to component name ',()=>{

var fixture = TestBed.createComponent(UserComponent);

var app = fixture.debugElement.componentInstance;

var userService = fixture.debugElement.injector.get(UserService);

fixture.detectChanges();

expect(userService.user.name).toEqual(app.user.name)

}

});

it('Sum should be 4',()=>{

var fixture = TestBed.createComponent(UserComponent);

var app = fixture.debugElement.componentInstance;

var calculatorService = fixture.debugElement.injector.get(CalculatorService);

fixture.detectChanges();

expect(calculatorService.add(2,2)).toEqual(4);

});

\*/

it('Should not fetch data if not called asynchronus way',fakeAsync(()=>{

var fixture = TestBed.createComponent(UserComponent);

var app = fixture.debugElement.componentInstance;

var dataService = fixture.debugElement.injector.get(DataService);

let spy = spyOn(dataService,'getData').and.returnValue(Promise.resolve('Data'));

fixture.detectChanges();

tick();

expect(app.sRecord).toBe('Data');

}));

});

--

**Isolated and non-isolated tests :**

**Test Pipe :**