Unit Test Services :

<https://dzone.com/articles/angular-2-testing-in-depth-services>

First of create **horse.service.ts**

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

@Injectable()

export class Engine {

getHorsepower() {

return 150;

}

getName() {

return 'Basic engine';

}

}

Now go to your speck file and import the service file there :

import {Engine} from './services/horse.service';

Now apply the it block test there inside the describe section :

it('should return it\'s horsepower', () => {

let subject = new Engine();

expect(subject.getHorsepower()).toEqual(150);

});

Now let us create a Calculator service and then test it’s functions :

Calculator.service.ts

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

@Injectable()

export class CalculatorService

{

mul(t1:number,t2:number){

return (t1\*t2)

}

add(a:number,b:number){

return (a+b);

}

}

Now lets create spec file and write it’s it block inside the describe block :

it('Multiply test',()=>{

let calculator = new CalculatorService();

expect(calculator.mul(5,3)).toEqual(15);

});

it('Add test',()=>{

let calculator = new CalculatorService();

expect(calculator.add(2,3)).toEqual(5);

});

**Note : we have initialized the CalculatorService object in two it blocks. Better to initialize it inside the beforeEach() block.**

let calculator : CalculatorService;

beforeEach(async(() => {

calculator = new CalculatorService();

}

**Using the dependency injection :**

Create a service triangle.service.ts

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

import {Perimeter} from './perimeter.service';

@Injectable()

export class TriangleService{

constructor(private perimeter : Perimeter){}

area(a:number,b:number,c:number){

let p = this.perimeter.peri(a,b,c);

let ar = Math.sqrt(p\*(p-a)\*(p-b)\*(p-c));

return ar;

}

}

**It is dependent on Perimeter service :**

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

@Injectable()

export class Perimeter

{

peri(a:number,b:number,c:number){

return (a+b+c)/2.0;

}

}

**Inside the speck file import both the servives :**

import {Perimeter} from './services/perimeter.service';

import {TriangleService} from './services/triangle.service';

**Declare providers in dependency module :**

let tri : TriangleService;

beforeEach(() => {

TestBed.configureTestingModule({

providers: [Perimeter, TriangleService]

});

});

**Now inject it :**

beforeEach(inject([TriangleService], (triangleService: TriangleService) => {

tri = triangleService;

}));

**Now at last apply the it block :**

it('Triangle Service',()=>{

let ans = tri.area(2,3,2);

console.log('The area is '+ans)

expect(tri.area(2,3,2)).toEqual(1.984313483298443);

});

**Now , let’s see new example . Here we will calculate the Gross salary from the basic salary :**

Create the SalaryService file : **salary.service.ts**

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

import {Hra} from './hra.service';

import {Da} from './da.service';

@Injectable()

export class SalaryService{

constructor(private daService : Da,private hraService : Hra){

}

grossSalary(bs){

let hra = this.hraService.getHra(bs);

let da = this.daService.getDa(bs);

let gross = bs+hra+da;

return gross;

}

}

**Again create the da.service.ts :**

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

@Injectable()

export class Da{

getDa(bs : number){

let da = (bs\*11)/100;

return da;

}

}

**Also create hra.service.ts**

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

@Injectable()

export class Hra{

getHra(bs : number){

let hra = (bs\*30)/100;

return hra;

}

}

**Now create the spect file : Import all the dependencies first :**

import {Hra} from './services/hra.service';

import {Da} from './services/da.service';

import {SalaryService} from './services/salary.service';

**Then Configure the providers :**

beforeEach(()=>{

TestBed.configureTestingModule({

providers : [Da,Hra,SalaryService]

});

});

**Then inject the service :**

beforeEach(inject([SalaryService],(salaryService : SalaryService)=>{

salSer = salaryService;

})

**At last create the it block :**

it('SalaryTest',()=>{

var sal = salSer.grossSalary(12000);

expect(sal).toEqual(16920);

});