**T10-HOL001\_TestCase [Testing Angular Service]**

In this hands on we will test Angular Service [stand alone service testing as well as testing the service injected under a component].

Tasks:

1. Create a service class called Calculation. Under the class create a method called BiilCalculation() based on the quantity, perProductCost and gst amount.

**Sample Code is given for your reference:**

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

@Injectable({

  providedIn: 'root'

})

export class CalculationService {

  totalCost:number;

  gstAmount:number;

  billAmount:number;

  BiilCalculation(quantity:number,perProductCost:number)

  {

    this.totalCost=quantity\*perProductCost;

    this.gstAmount=(this.totalCost\*13.5)/100;

    this.billAmount=this.totalCost+this.gstAmount;

    return this.billAmount;

  }

}

1. Under calculation.service.spec.ts inject the CalculationService [to perform stand alone service testing] and check if the result given by BillCalculation() method is equal to expected result.

**Sample Code is given for your reference:**

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

import { CalculationService } from './calculation.service';

describe('CalculationsService tests', () => {

    let calculationsSvc: CalculationService;

    beforeEach(inject(

      [CalculationService],

      (calcService: CalculationService) => {

        calculationsSvc = calcService;

      }

    ));

  it("should calcuate Total Bill Based on Quantity\*Price+13.5%GST Amount", () => {

    let result = calculationsSvc.BiilCalculation(5, 4500);

    expect(result).toEqual(25537.5);

  });

  });

***Note: Here we didn’t add the TestBed setup here. We didn’t do this as CalculationService is provided in the root injector. Otherwise, if the service is provided in a module or in a component, we need to provide the service in the testing module configured with TestBed. The beforeEach block gets object of the service from the root injector. Now this object can be used to call the add method and test it.***

1. Under the constructor of AppComponent class inject the CalculationService. Call the BillAmount() method of service class and calculate the bill amount.

**Sample Code is given for your reference:**

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

import {CalculationService} from './calculation.service';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'Bill Calculation';

  quantity:number=3;

  perProductCost:number=1000;

  totalBillAmount:number;

  constructor(private calculationService:CalculationService){

  this.totalBillAmount=this.calculationService.BiilCalculation(this.quantity,this.perProductCost);

  }

}

1. Under app.component.html display the AppComponent title, Quantity, Per Product Cost and Bill Amount.

**Sample Code is given for your reference:**

<h1>{{title}}</h1>

<b>Quantity: {{quantity}}</b>

<br/>

<b>Per Product Cost: {{perProductCost}}</b>

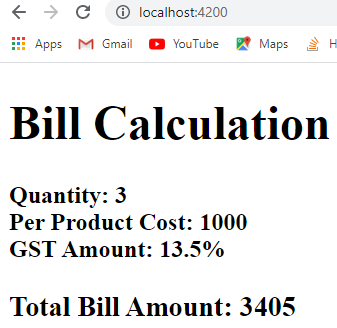
<br/>

<b>GST Amount: 13.5%</b>

<br/>

<h3>Total Bill Amount: {{totalBillAmount}}</h3>

**Sample Output is given for your reference:**

****

1. Check if the describe () block for AppComponent is present under app.component.spec.ts.
2. Check whether a beforeEach() block that instantiates a new instance of AppComponent for each spec is present. Also under beforeEach() block add providers:[CalculationService]
3. Check if there is a test case to check whether the AppComponent is created is present.
4. Create a test case to check if the title with value “Bill Calculation” is present.
5. Create a test case to check if the title is displayed in an h1 tag.
6. Create a test case to check if the result given by calling the BillAmount() method of service class is equal to expected output.
7. Create a test case to check if the bill amount is rendered in an h3 tag.

**Sample Code is given for your reference:**

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

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

import { CalculationService } from './calculation.service';

describe('AppComponent', () => {

  beforeEach(async(() => {

    TestBed.configureTestingModule({

      declarations: [

        AppComponent

      ],

      providers:[CalculationService]

    }).compileComponents();

  }));

  it('should create the AppComponent', () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.debugElement.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'Bill Calculation'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.debugElement.componentInstance;

    expect(app.title).toEqual('Bill Calculation');

  });

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

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.debugElement.nativeElement;

    expect(compiled.querySelector('h1').textContent).toContain('Bill Calculation');

  });

  it('should calculate bill amount', async(() => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.debugElement.componentInstance;

    let service=new CalculationService();

    app.quantity=6;

    app.perProductCost=2500;

    expect(service.BiilCalculation(app.quantity,app.perProductCost)).toEqual(17025);

  }));

  it('should render bill amount in a h3 tag', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.debugElement.nativeElement;

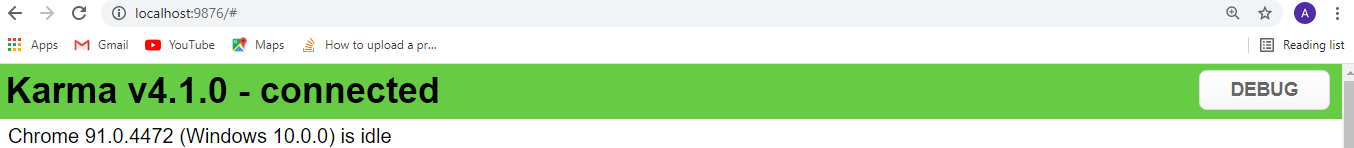
    expect(compiled.querySelector('h3').textContent).toContain('Total Bill Amount: 3405');

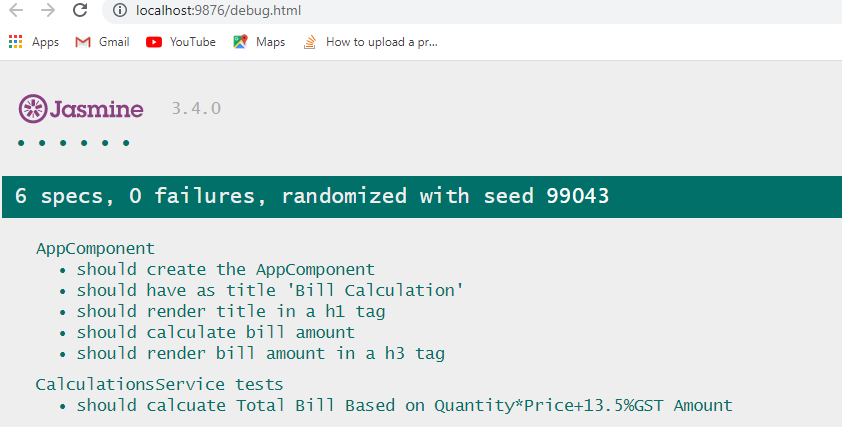
  });

});

1. Run the command ng test in cli and click the debug button of Karma window in browser to see the jasmine test results.

**Sample output is given for your reference:**

****

****