**T15-HOL001\_TestCase [Unit testing and e2e testing of Angular Application]**

In this hand’s on, we will perform unit testing on angular services using Jasmine and Karma and end-to-end testing of the application using Protractor.

**Tasks:**

1. Create an application called **T15-HOL001**
2. Run the command: **npm install screenshot-protractor –save.** This is used to install protractor screenshot to capture Protractor test output.
3. Run the command: **npm i bootstrap** to install bootstrap for this application.
4. Open style.css file and add the path of bootstrap.

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

@import "~bootstrap/dist/css/bootstrap.min.css";

1. User xdescribe() block under **app.component.spec.ts**
2. Under the application create a service class called ProductList (use the command: **ng g service productlist**)
3. Under the service class do the following:
4. Create an array called products with some default values
5. Create a method called addProduct(product: string) to add a new product to the products array
6. Create a method called getProducts() to return all data present under products array.

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

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

@Injectable({

  providedIn: 'root'

})

export class ProductlistService {

  products=['AC','Fridge','TV'];

  addProduct(product: string) {

    if (!(this.products.filter(u => u === product).length > 0)) {

      this.products.push(product);

    }

  }

  getProducts() {

    return this.products;

  }

}

1. Under **productlist.service.spec.ts** file do the following to unit test your service class:
2. Under beforeEach() block inject the ProductListService
3. Create a test case to check if the last element in the products array is the newly added data that you have inserted under the array.

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

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

import { ProductlistService } from './productlist.service';

describe('ProductlistService', () => {

  let productListSvc: ProductlistService;

    beforeEach(inject(

      [ProductlistService],

      (productListService: ProductlistService) => {

      productListSvc = productListService;

      }

    ));

  it("should add product and display the product added", () => {

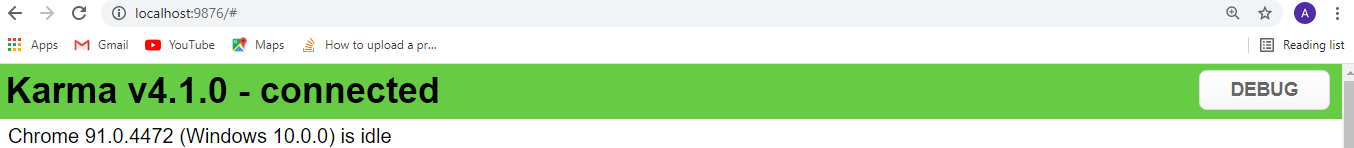
    productListSvc.addProduct('Ear Bud');

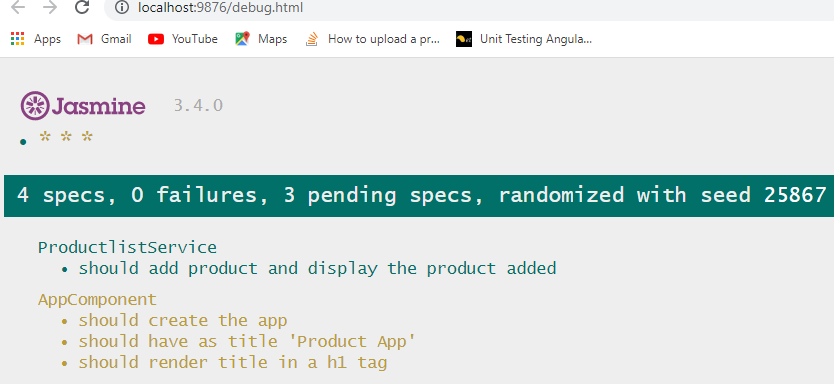
    expect(productListSvc.getProducts().pop()).toEqual('Ear Bud');

  });

});

**Sample output of unit test on service class is given for your reference which we get by running ng test command and clicking on the Debug button on Karma window in the browser:**

****

****

1. Under the **AppComponent** class do the following:
2. Under the constructor, inject the ProductListService.
3. Create a variable called title with a value “Product App”.
4. Create a method called addProduct(). Under the method call the addProduct() method from service class to add data to array.
5. Create a method called getProducts(). Under the method call the getProducts() method from service class to display the products present under the array.

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

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

import{ProductlistService} from './productlist.service';

import{FormsModule} from '@angular/forms';

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

Data to be binded with .html input control

export class AppComponent {

  title = 'Product App';

  newProduct='';

  constructor(private productListService:ProductlistService) { }

  addProduct() {

    this.productListService.addProduct(this.newProduct);

  }

  getProducts() {

    return this.productListService.getProducts();

  }

}

1. Under **app.component.html** do the following:
2. Display the title under an h1 tag
3. Create an input control and bind it to ngModel directive
4. Create a button control and under its click event call the addProduct() method
5. Create an unordered list and display the product data from products array by calling the getProducts() method under ngFor directive.

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

<div class="container">

  <h1>

    Welcome to {{ title }}!

  </h1>

 <br/><br/>

 <input id="newProduct" [(ngModel)]="newProduct" calss="ng-form"><br/><br/>

 <button id="addProductBtn" (click)="addProduct()" class="btn btn-success">Add Product</button><br/>

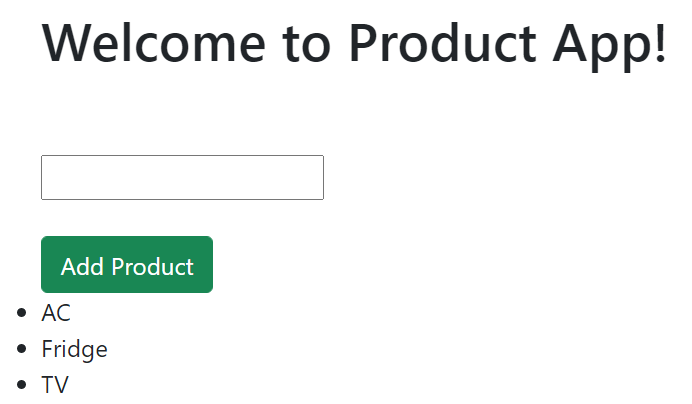
    <ul class="list-group">

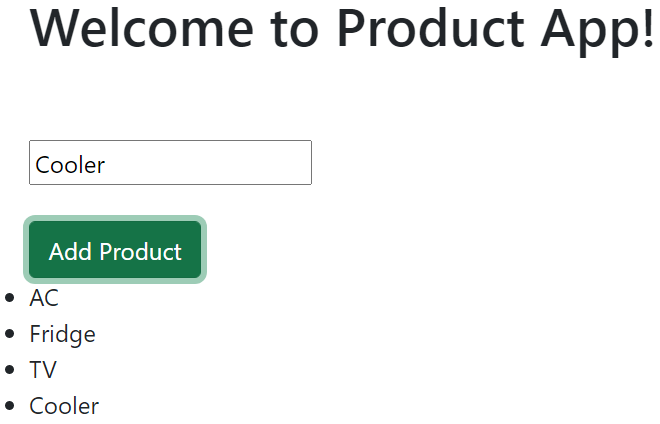
      <li \*ngFor="let product of getProducts()"> {{ product }} </li>

    </ul>

</div>

**Sample output is given for your reference which we get by running ng s command:**

****



1. Now for e2e testing, expand the e2e/src folder present under the application.
2. Under **app.po.ts** file do the following:
3. Create a method called setNewProduct(product: string). This method will find the input element present under app.component.html file by its id and send its data under product passed as method argument.
4. Create a method called clickAddProduct() which will find the button control present under app.component.html by its id and get its click event.
5. Create a method called getProductsList() which will find the display list present under app.component.html using the css class name.

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

import { browser, by, element } from 'protractor';

export class AppPage {

  navigateTo() {

    return browser.get(browser.baseUrl) as Promise<any>;

  }

  getTitleText() {

    return element(by.css('app-root h1')).getText() as Promise<string>;

  }

  setNewProduct(product: string) {

    element(by.id('newProduct')).sendKeys(product);

  }

  clickAddProduct() {

    element(by.id('addProductBtn')).click();

  }

  getProductsList() {

    return element.all(by.css('.list-group li')).last().getText();

  }

}

1. Under app**.e2e-spec.ts** file do the following:
2. Under afterEach() block write your coding to capture Protractor test output.
3. Modify the test case to display the welcome message present under title which should be equal to ‘Welcome to Product App!’.
4. Create a test case to check if the newly added product data is displayed under the list.

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

import { AppPage } from './app.po';

import { browser, logging } from 'protractor';

import { createWriteStream } from 'fs';

describe('workspace-project App', () => {

  let page: AppPage;

  beforeEach(() => {

    page = new AppPage();

  });

  afterEach(() => {

    browser.takeScreenshot().then((png) =>{

    var stream = createWriteStream("samplescreenshot.png"); /\*\* change the png file name as per your choice  \*/

    stream.write(new Buffer(png, 'base64'));

    stream.end;

    });

    });

  it('should display welcome message', () => {

    page.navigateTo();

    expect(page.getTitleText()).toEqual('Welcome to Product App!');

  });

  it('product addition', () => {

    page.navigateTo();

    page.setNewProduct('Earphone');

    page.clickAddProduct();

    expect(page.getProductsList()).toEqual('Earphone');

  });

  afterEach(async () => {

    // Assert that there are no errors emitted from the browser

    const logs = await browser.manage().logs().get(logging.Type.BROWSER);

    expect(logs).not.toContain(jasmine.objectContaining({

      level: logging.Level.SEVERE,

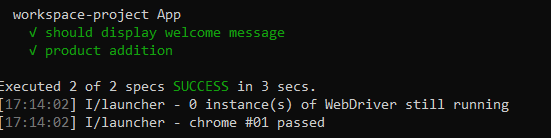
    } as logging.Entry));

  });

});

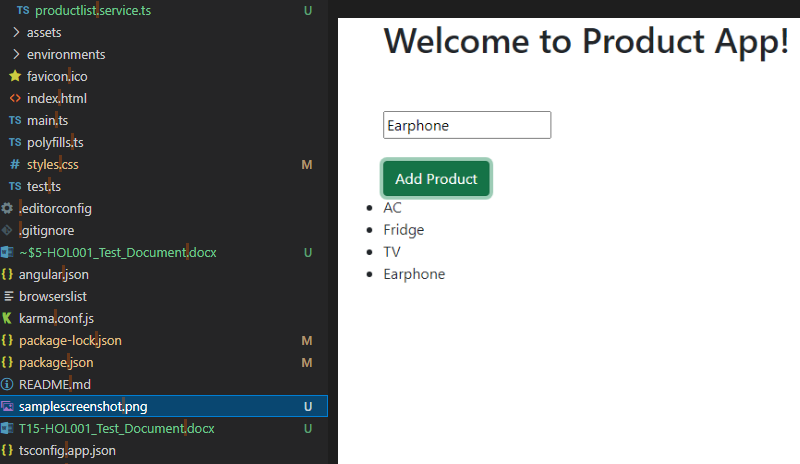
1. Execute the e2e testing by using the command: **ng e2e –port=4300** [Note: You can change the port number according to your choice].

**Sample Test Result under command window:**

****

1. Finally go under file explorer of your application and check if the Protractor test result has been captured.

**Sample of Protractor test output capture is given for your reference:**

****