**Angular-T07-HOL001\_TestCase [Template Drive Form Testing]**

In this hands on we will understand how to perform unit testing of Angular applications using Jasmine and Karma on Template Drive Form Testing.

Tasks:

1. Create a Component called TraineeComponent. Add the below code in the component class:

import { Component, OnInit, ViewChild } from '@angular/core';

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

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

@Component({

  selector: 'app-trainee',

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

  styleUrls: ['./trainee.component.css']

})

export class TraineeComponent implements OnInit {

 @ViewChild('registerForm',{static:true}) registerForm:NgForm;

 message:string;

 traineename:string;

 emailid:string;

 constructor() { }

 ngOnInit() {

   this.message='';

 }

  public submitForm()

  {

    if(this.registerForm.valid)

    {

      this.message='Registration Form Is Valid';

      this.traineename='';

      this.emailid='';

      this.registerForm.reset();

    }

  }

}

1. On the trainee.component.html file create a form with traineename and emailed fields. Also both the input fields should contain proper validation information.
2. Create a submit button which should be disabled if the form is invalid. If the form is valid (i.e. the input fields are valid); a success message should be displayed on button click.

**Sample code of html file is given for your reference:**

<div class="container">

<form name="registerForm" #registerForm="ngForm" action="">

    <div class="row">

        <div class="col-md-12">

            <h2>Trainee Registration Form</h2>

        </div>

    </div>

    <br/>

    <div class="row">

        <div class="col-md-12">

            <span id="success-msg" \*ngIf="message!==''"  class="alert alert-success" >{{message}}</span>

        </div>

    </div>

    <br/>

    <div class="row">

        <div class="col-md-12">

            <div class="form-group">

                <label for="traineename">Enter Trainee Name:</label>

                <input required #traineename="ngModel" name="traineename" ngModel type="text" class="form-control">

                <br/>

                <span class="alert alert-danger" \*ngIf="traineename.touched && traineename.invalid">\*Invalid Name</span>

            </div>

        </div>

    </div>

    <div class="row">

        <div class="col-md-12">

            <div class="form-group">

                <label for="emailid">Enter Email Id:</label>

                <input required email #emailid="ngModel" name="emailid" ngModel type="text" class="form-control">

                <br/>

                <span class="alert alert-danger" \*ngIf="emailid.touched && emailid.invalid && emailid.errors['email']">\*Invalid Email Id</span>

            </div>

        </div>

    </div>

    <div class="row">

        <div class="col-md-12">

           <button class="btn btn-success" (click)="submitForm()" [disabled]="registerForm.invalid" id="submitButton">Register</button>

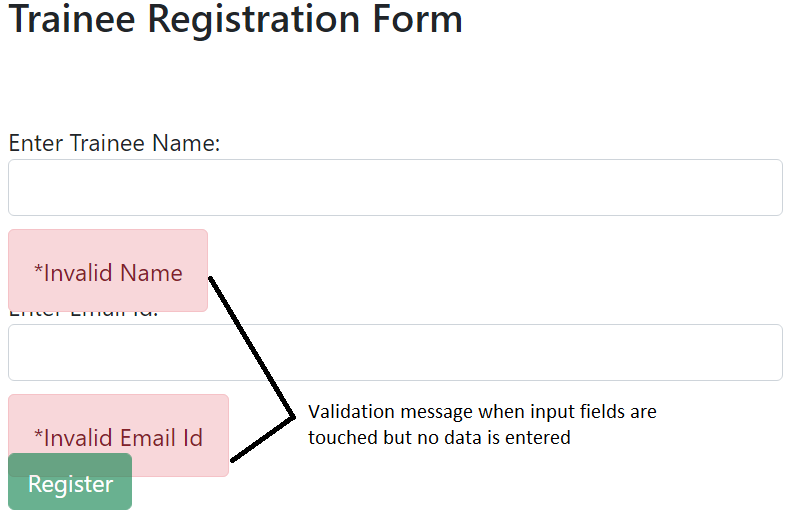
        </div>

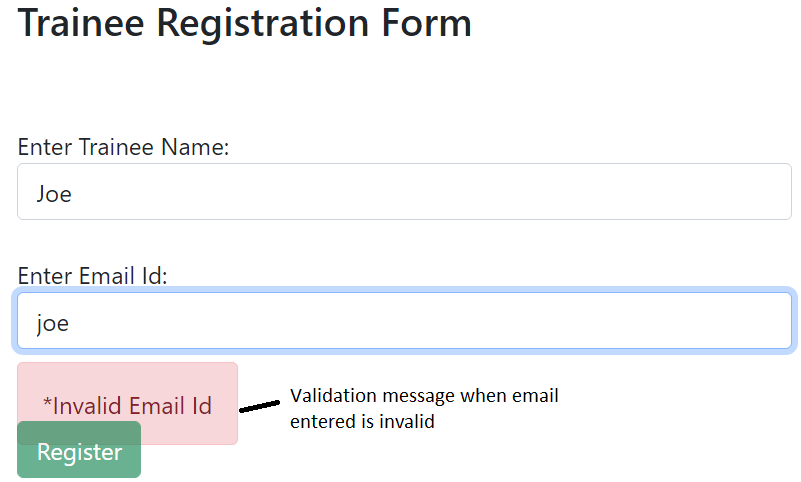
    </div>

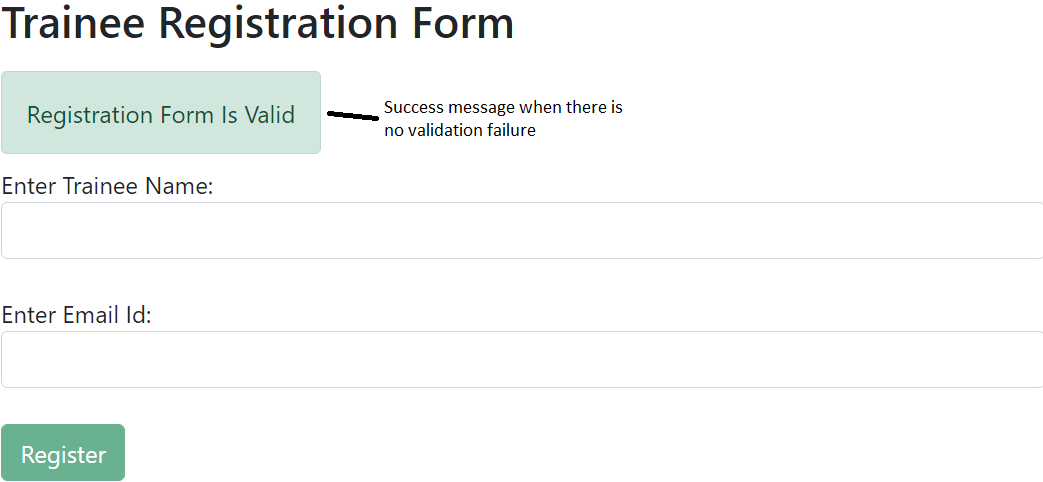
</form>

    </div>

**Sample output is given below:**







1. Use xdescribe() for app.component.spec.ts
2. Check whether the file trainee.component.spec.ts is present in src/app/trainee.
3. Check if the describe () block for TraineeComponent is present.
4. Check whether a beforeEach() block that instantiates a new instance of TraineeComponent for each spec is present. Also under beforeeach() block add imports:[FormsModule] as we are working with template driven form and ngModel directive.
5. Check if there is a test case to check whether the TraineeComponent is created is present.
6. Create a test case to checkout if the traineename input field and registerForm validation fails if the input field is touched but no data is entered by user.
7. Create a test case to checkout if the emailid input field and registerForm validation fails if the input field is touched but no data is entered by user.
8. Create a test case to checkout if the emailid input field and registerForm validation fails if invalid email id is entered by user.
9. Create a test case to check if success message is displayed in the span with id=”success-msg” if the input fields are valid and register form is valid.

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

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

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

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

import { TraineeComponent } from './trainee.component';

describe('TraineeComponent', () => {

  let component: TraineeComponent;

  let fixture: ComponentFixture<TraineeComponent>;

  beforeEach(async(() => {

    TestBed.configureTestingModule({

      declarations: [ TraineeComponent ],

      imports:[FormsModule]

    })

    .compileComponents();

  }));

  beforeEach(() => {

    fixture = TestBed.createComponent(TraineeComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create TraineeComponent', () => {

    expect(component).toBeTruthy();

  });

  it('Trainee Name - Check | No Data Supplied But Input Field is Touched',async()=>

  {

    fixture.whenStable().then(()=>{

    let traineename=component.registerForm.form.controls['traineename'];

    traineename.setValue('');

    expect(traineename.touched).toBeFalsy();

    expect(traineename.valid).toBeFalsy();

    expect(component.registerForm.valid).toBeFalsy();

    });

  });

  it('Enmail Id - Check | No Data Supplied But Input Field is Touched',async()=>

  {

    fixture.whenStable().then(()=>{

      let emailid=component.registerForm.form.controls['emailid'];

      emailid.setValue('');

      expect(emailid.touched).toBeFalsy();

      expect(emailid.valid).toBeFalsy();

      expect(component.registerForm.valid).toBeFalsy();

    });

  });

  it('Email Id - Check | Invalid Email Id Supplied',async()=>

  {

    fixture.whenStable().then(()=>{

    let emailid=component.registerForm.form.controls['emailid'];

    emailid.setValue('abc');

    expect(emailid.errors['email']).toBeTruthy();

    expect(emailid.valid).toBeFalsy();

    expect(component.registerForm.valid).toBeFalsy();

    });

  });

    it('When Both Trainee Name and Email Is Valid, Form Should be submitted Successfully',async()=>{

      fixture.whenStable().then(()=>{

        let traineename=component.registerForm.form.controls['traineename'];

        let emailid=component.registerForm.form.controls['emailid'];

        traineename.setValue('Sam');

        emailid.setValue('sam@gmail.com');

        component.submitForm();

        fixture.detectChanges();

        let successMsg=fixture.debugElement.query(By.css('#success-msg'));

        expect(successMsg.nativeElement.innerText).toBe(component.message);

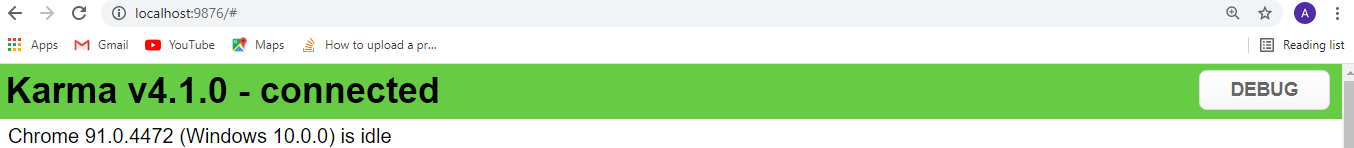
      });

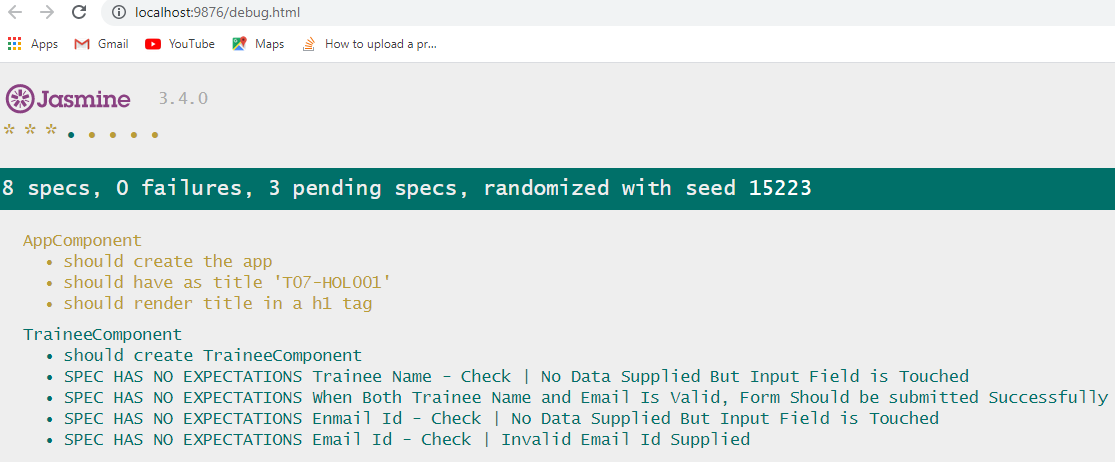
    });

});

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:**

****

****