**Angular-T05-HOL001\_TestCase [Testing Custom Pipes]**

In this hands on we will understand how to perform unit testing of Angular applications using Jasmine and Karma on custom pipes.

**Tasks:**

1. Create a custom pipe called Phonenumber [use the command ng g pipe Phonenumber]. This pipe is used to format land line number. Code is given below for your reference:

import { Pipe, PipeTransform } from '@angular/core';

@Pipe({

name: 'phonenumber'

})

export class PhonenumberPipe implements PipeTransform {

transform(value: any,args?: any[]): any {

if (undefined !== value && value.length === 10) {

     return '(' + value.substring(0, 3) + ') ' + value.substring(3, 6) + ' ' + value.substring(6);

 }

else{

 return 'Invalid Landline Number';

 }

}

1. Check if the file phonenumber.pipe.spec.ts is present
2. Check if the describe () block for PhonenumberPipe is present.
3. Check if there is a test case to check whether the PhonenumberPipe is created is present.
4. Create a test case to check if the landline number is formatted according to the format (xxx) xxx xxxx.
5. Create a test case to check if the message ‘Invalid Landline Number’ is given if landline number is given less than 10 digits.

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

import { PhonenumberPipe } from './phonenumber.pipe';

describe('PhonenumberPipe', () => {

  it('create an instance', () => {

    const pipe = new PhonenumberPipe();

    expect(pipe).toBeTruthy();

  });

  it('should display in phone format', () => {

    const phoneNumber = '3333333333';

    const pipe = new PhonenumberPipe();

    const result = pipe.transform(phoneNumber);

    expect(result).toBe('(333) 333 3333');

});

it('should display Invalid Phone Number if phone number is invalid', () => {

  const phoneNumber = '333333';

  const pipe = new PhonenumberPipe();

  const result = pipe.transform(phoneNumber);

  expect(result).toBe('Invalid Landline Number');

});

});

1. Create a component called Employee. Under this class add the following data:

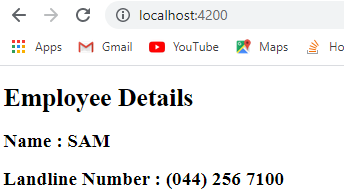
title="employee details";

name="sam";

landlineNumber="0442567100";

1. Under employee.component.html file, add pipes to get the following display:
2. The title should be in title case and present under h1 tag (use titlecase pipe)
3. The name should be in upper case and present in a h2 tag with id=”name” (use uppercase pipe)
4. The landline number should be displayed as (044) 256 7100 (use the phonenumber pipe) under an h2 tag with id=”phoneNumber”.

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



1. Use xdescribe() for app.component.spec.ts
2. Check whether the file employee.component.spec.ts is present in src/app/employee.
3. Check if the describe () block for EmployeeComponent is present.
4. Check whether a beforeEach() block that instantiates a new instance of EmployeeComponent for each spec is present
5. Check if there is a test case to check whether the EmployeeComponent is created is present.
6. Create a test case to check whether there is a title called “employee details” is present.
7. Create a test case to check whether there is a name called “sam” is present.
8. Create a test case to check whether there is landline number with value “0442567100” is present.
9. Create a test case to check if the title is present in h1 tag as “Employee Details”
10. Create a test case to check if the name is present in h2 tag with id=”name” as “Name: SAM”
11. Create a test case to check if the landline number is present in h2 tag with id=” phoneNumber” as “(044) 256 7100”
12. Run ng test to test your application. Sample output is given for your reference:

