**Two-Way Binding**

**Child.component.html**

<h1>child component</h1>

<br>

<h2>Msg sendig to parent component</h2>

<input type="text" #ctext

(keyup)="onchange(ctext.value)">

<br>

<p> Value coming from parent component : <b> {{pdata}} </b></p>

**Child.component.specs.ts**

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

import { ChildComponent } from './child.component';

describe('ChildComponent', () => {

  let component: ChildComponent;

  let fixture: ComponentFixture<ChildComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ ChildComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(ChildComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Child.component.ts**

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

@Component({

  selector: 'app-child',

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

  styleUrls: ['./child.component.css'],

  inputs:['pdata'],

  outputs:['cevent'],

})

export class ChildComponent implements OnInit {

  pdata:string="";

  constructor() { }

  ngOnInit(): void {

  }

  cevent=new EventEmitter<string>();

  onchange(value:string){

    this.cevent.emit(value);

  }

}

**App-routing.module.ts**

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

import { RouterModule, Routes } from '@angular/router';

const routes: Routes = [];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

**app.component.html**

<h1>Parent Component</h1>

<input type="text" (keyup)="0" #ptext/>

<h3>Value coming from child component is : <b>{{cdata}}</b></h3>

<app-child [pdata]="ptext.value" (cevent)="cdata=$event"></app-child>

**App.component.specs.ts**

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

import { RouterTestingModule } from '@angular/router/testing';

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

describe('AppComponent', () => {

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      imports: [

        RouterTestingModule

      ],

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  });

  it('should create the app', () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app).toBeTruthy();

  });

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

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app.title).toEqual('parent');

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement as HTMLElement;

    expect(compiled.querySelector('.content span')?.textContent).toContain('parent app is running!');

  });

});

**App.component.ts**

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'parent';

  cdata:string="";

}

**App.module.ts**

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

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

import { AppRoutingModule } from './app-routing.module';

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

import { ChildComponent } from './child/child.component';

@NgModule({

  declarations: [

    AppComponent,

    ChildComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

**index.html**

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Parent</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

  <app-root></app-root>

</body>

</html>