Component Level Interactions

Create App> ng new complevelinter

>cd complevelinter

>npm start / ng server

Open the app in VC Code.

For Communication:

Parent to child

Ng g c child

Open app.component.html

<div class="container">

<h1>Parent Component</h1>

Enter Message <input type="text" #ptext

(keyup)="0" />

<!--Pass Data from Parent to child-->

<app-child [pdata]="ptext.value"></app-child>

</div>

Child.component.ts

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

@Component({

selector: 'app-child',

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

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

inputs:['pdata']

})

export class ChildComponent {

pdata:string="";

}

Child.component.html

<h3>Child Component</h3>

<h5>Value Coming from Parent Component {{pdata}}</h5>

Child to Parent Communication

Go to Child.component.ts

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

@Component({

selector: 'app-child',

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

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

inputs:['pdata'],

outputs:['cevent']

})

export class ChildComponent {

pdata:string="";

//Event emitter object for emitting event

cevent:EventEmitter<string>= new EventEmitter<string>();

//function

onchange(value:string){

this.cevent.emit(value); //emit the received value

}

}

Child.component.html

<h3>Child Component</h3>

<input type="text" #ctext (keyup)="onchange(ctext.value)" />

<h5>Value Coming from Parent Component {{pdata}}</h5>

App.component.ts

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'complevelinter';

cdata:string="";

}

App.component.html

<div class="container">

<h1>Parent Component</h1>

<h3>Value Coming from Child Component: {{cdata}}</h3>

Enter Message <input type="text" #ptext

(keyup)="0" />

<!--Pass Data from Parent to child-->

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

</div>

Now check communication between parent and child.