

Input and Output

Input and Output are two decorators in Angular responsible for communication between two components.

Input

In a component needs to receive data from another component in other words parent component, we need to decorate the variable which receives the values as input.

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

@Component({
  selector: 'app-child',
  templateUrl: './child.component.html',
  styleUrls: ['./child.component.css']
})
export class ChildComponent implements OnInit {

  @Input('info') data:string="Child Component ....";

  constructor() { }

  ngOnInit() {
  }

}
```

In the above example, the variable data is decorated with @Input, and in the template

We print data

```
1 <div class="unit1">
2   {{data}}
3 </div>
4
```

In the parent component we pass the data child as follows

The class file of parent component appears as follows

```
module.ts # child.component.css <> app.component.html TS filternames.pipe.ts TS app.component.ts x
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'app';
10  developers=["Raghav","Ravi","Ramu","Girish","Timur","Bhoomi",
11  "Chirst","Lord","Manish","Vinay","Vijay","Pathan","Balu","Baggy"];
12
13
14 }
15
```

And we pass the value to the child in the following manner

```
module.ts # child.component.css <> app.component.html x TS filternames.pipe.ts TS app.component.ts
1 Search name <input type="text" [(ngModel)]="pattern">
2 <br>
3 <b><u>Names</u></b>
4 <br>
5 <app-child [info]="x" *ngFor="let x of developers"></app-child>
6
```

It yields the following output

Search name **Names**

Raghav

Ravi

Ramu

Girish

Timur

Bhoomi

Chirst

Lord

Manish

Output

Angular is based on a one-directional data flow and does not have two-way data binding. So, how do you get a component to emit an event to another component?

In Angular, a component can emit an event using **@Output** and **EventEmitter**. Both are parts of the **@angular/core**.

If a Child wants to communicate something the parent then we need to have a event emitter in the child declared as follows

```

1  import { Component, OnInit, Input, Output,
2  EventEmitter } from '@angular/core';
3
4  @Component({
5    selector: 'app-child',
6    templateUrl: './child.component.html',
7    styleUrls: ['./child.component.css']
8  })
9  export class ChildComponent implements OnInit {
10
11    @Input('info') data:string="Child Component ....";
12    @Output('emit') emitInfo:EventEmitter<string>
13    =new EventEmitter<string>();
14    constructor() { }
15
16    ngOnInit() {
17    }
18
19    postData(){
20      this.emitInfo.emit(this.data);
21    }
22  }

```

In the above code, we have a variable called emitInfo which is of type Event Emitter and when the function postData is called, it emits data

Lets see how the postData function is called. In the template,

```

<div class="unit1" (click)="postData()">
  {{data}}
</div>

```

When the child template is clicked, we call post data

To subscribe to this even emitter, we do following changes in parent

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app';
  developers=["Raghav","Ravi","Ramu","Girish","Timur","Bhoomi",
"Chirst","Lord","Manish","Vinay","Vijay","Pathan","Balu","Baggy"];

  consumeData(data){
    this.title="Data recieved from child"+data;
  }
}
```

In the template , you can see



```
1 <h1>{{title}}</h1>
2 Search name <input type="text" [(ngModel)]="pattern">
3 <br>
4 <b><u>Names</u></b>
5 <br>
6 <app-child [info]="x" (emit)="consumeData($event)"
7 *ngFor="let x of developers"></app-child>
```

We have subscribe to the event called emit using output directive notation. When we click division with name girish, it changes the title in the parent

Data recieved from childGirish

Search name

Names

Raghav

Ravi

Ramu

Girish

Timur

Bhoomi

Chirst

Lord

Manish

Vinay