

Parent-Child Component Interaction

A Component in Angular can have child components. Also, those child components can have their own further child components. Angular seamlessly supports nested components.

Parent & Child components can interact with each other by exchanging data. The data can be of any type.

Parent to child interaction

Let us consider AppComponent has a child called ChildComponent. Then, the selector of ChildComponent ('app-child') was placed in the view of AppComponent (app.component.html)

By using that selector, the parent component can send data to its children. To receive data from parent as input, the child component should have an instance variable which is exposed to parent by adding @Input() decorator to it.

Child.component.ts:

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

@Component({.....})

Export class ChildComponent

{

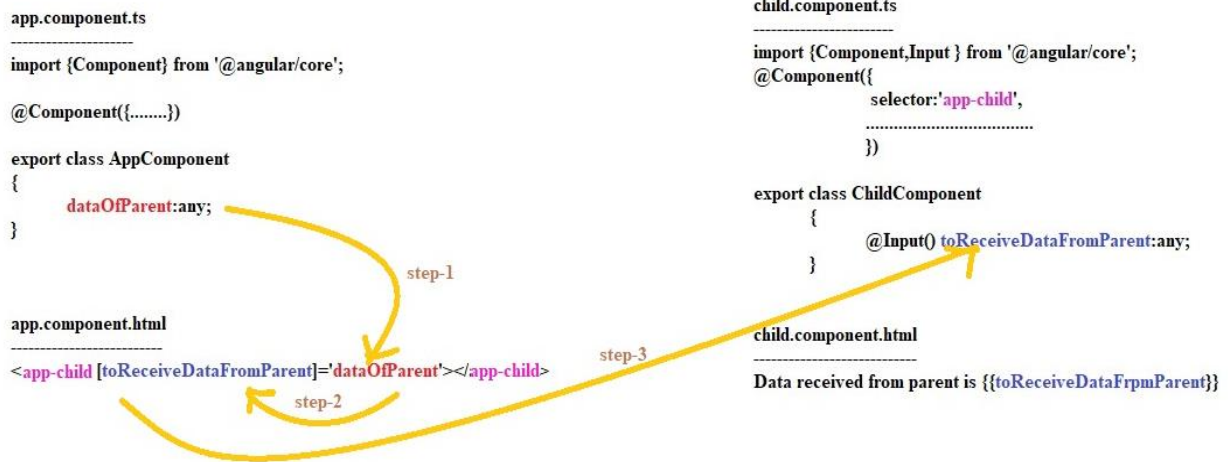
    @Input() toReceiveDataFromParent:any;

}
```

Then parent component can send data to its children using property binding at selector.

App.component.html:

```
<app-child [toReceiveDataFromParent]="dataOfParent"></app-child>
```



Child to parent interaction

Parent component transfers data to child component by using selector of child component. But when child component need to transfer data to parent component, it not possible the same way, because parent component's selector is not available to child component. So that, by using following steps, it can be done

Step-1: create a custom event in child component and apply `@Output()` decorator to it.

To create custom event, create object for `EventEmitter` class.

```
@Output() myEvent=new EventEmitter();
```

Step-2: let us assign job to this custom event by calling `emit(data)` on it.

```
myEvent.emit(dataOfChild);
```

Whenever, this statement is executed, then it emits data to its parent.

Step-3: Parent component can handle the custom event of child component to receive the value

emitted by it. The data emitted by an event is available in `“$event”` object.

```
<app-child (myEvent)="toReceiveDataFromChild=$event"></app-child>
```

So, whenever the event is occurred, the emitted value of it will be assigned to variable of parent component.

App.component.ts:

```

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

@Component({.....})

Export class AppComponent
{
    toReceiveDataFromChild:any;
}

```

App.component.html:

```

<app-child (myEvent)="
    toReceiveDataFromChild=$event">

</app-child>

```

Child.component.ts:

```

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

@Component({.....})

export class ChildComponent
{
    dataOfChild:any;

    @Output() myEvent=new EventEmitter();

    sendDataToParent()
    {
        This.myEvent.emit(dataOfChild);
    }
}

```

Child.component.html:

```

<button (click)="sendDataToParent()>

    Send

</button>

```

