

## Lesson 02 Demo 01

### Component Level interactions

**Objective:** To setup the communication between the nested components using the @input and @output decorators.

**Tools required:** Ubuntu, Visual Studio, Node.js, and Angular

**Prerequisites:** Knowledge of JavaScript, Typescript, and Node.js

Steps to be followed:

1. Implementing the nested component that can communicate using the @input and @output decorators

#### Step 1: Implementing the nested component that can communicate using the @input and @output decorators:

- 1.1 Create the parent and child components using **ng g c parent** and **ng g c child** commands

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS  SQL CONSOLE

● Labuser@ubuntu2204:~/Desktop/Angular/myapp$ ng g c parent
CREATE src/app/parent/parent.component.css (0 bytes)
CREATE src/app/parent/parent.component.html (21 bytes)
CREATE src/app/parent/parent.component.spec.ts (599 bytes)
CREATE src/app/parent/parent.component.ts (202 bytes)
UPDATE src/app/app.module.ts (479 bytes)
● Labuser@ubuntu2204:~/Desktop/Angular/myapp$ ng g c child
CREATE src/app/child/child.component.css (0 bytes)
CREATE src/app/child/child.component.html (20 bytes)
CREATE src/app/child/child.component.spec.ts (592 bytes)
CREATE src/app/child/child.component.ts (198 bytes)
UPDATE src/app/app.module.ts (557 bytes)
○ Labuser@ubuntu2204:~/Desktop/Angular/myapp$
```

- 1.2 Copy the below code into the **parent.component.ts** file:

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

```
@Component({
  selector: 'app-parent',
  templateUrl: './parent.component.html',
  styleUrls: ['./parent.component.css']
})
export class ParentComponent {

  message = 'Hi from Parent Component'

  fromChild = '';
}
```



```
ts parent.component.ts U x
src > app > parent > ts parent.component.ts > ...
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-parent',
5    templateUrl: './parent.component.html',
6    styleUrls: ['./parent.component.css']
7  })
8  export class ParentComponent {
9
10     message = 'Hi from Parent Component'
11
12     fromChild = '';
13 }
```

1.3 Copy the below code into the **parent.component.html** file:

```
<h2>Parent Component</h2>
<h1>{{ fromChild }}</h1>
```

`<app-child [childData]="message" (customEvent)="fromChild=$event"></app-child>`

```

ts parent.component.ts U  parent.component.html U x
src > app > parent >  parent.component.html > ...
  Go to component
  1 <h2>Parent Component</h2>
  2 <h1>{{ fromChild }}</h1>
  3 <app-child [childData]="message" (customEvent)="fromChild=$event"></app-child>
  4

```

1.4 Copy the below code into the **child.component.ts** file:

```

import { Component, EventEmitter, Input, Output } from '@angular/core';
@Component({
  selector: 'app-child',
  templateUrl: './child.component.html',
  styleUrls: ['./child.component.css']
})
export class ChildComponent {

  @Input()
  childData = ''

  @Output()
  customEvent = new EventEmitter()

  onClickHandler(){
    this.customEvent.emit('Hi from Child')
  }
}

```

```

ts parent.component.ts U  <> parent.component.html U  ts child.component.ts U ●
src > app > child > ts child.component.ts > ...
1  import { Component, EventEmitter, Input, Output } from '@angular/core';
2
3  @Component({
4    selector: 'app-child',
5    templateUrl: './child.component.html',
6    styleUrls: ['./child.component.css']
7  })
8  export class ChildComponent {
9
10     @Input()
11     childData = ''
12
13     @Output()
14     customEvent = new EventEmitter()
15
16     onClickHandler(){
17       this.customEvent.emit('Hi from Child')
18     }
19   }

```

1.5 Copy the below code into the **child.component.html** file:

```

<h2>Child Component</h2>
<h1>{{ childData }}</h1>
<button (click)="onClickHandler()" >Send Data</button>

```

```

ts parent.component.ts U  <> parent.component.html U  ts child.component.ts U ●
src > app > child > <> child.component.html > ...
  Go to component
1  <h2>Child Component</h2>
2  <h1>{{ childData }}</h1>
3
4  <button (click)="onClickHandler()" >Send Data</button>
5

```

1.6 Copy and paste the below-mentioned code into the **app.component.html** file:

```

<div class="container">
  <app-parent></app-parent>
</div>

```

```

ts parent.component.ts U    <> parent.component.html U

src > app > <> app.component.html > ...
    Go to component | ...
1  <div class="container">
2  |   <app-parent></app-parent>
3  </div>
4  .

```

1.7 Start the server using the **ng s -o** command

```

labuser@ubuntu2204:~/Desktop/angular-project/myapp$ ng s -o
✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Raw Size
vendor.js           | vendor        | 2.04 MB
polyfills.js        | polyfills     | 314.26 kB
styles.css, styles.js | styles       | 209.39 kB
main.js             | main          | 7.67 kB
runtime.js          | runtime       | 6.51 kB
                    | Initial Total | 2.56 MB

Build at: 2023-02-01T18:53:58.507Z - Hash: d671567c6946c425 - Time: 9044ms

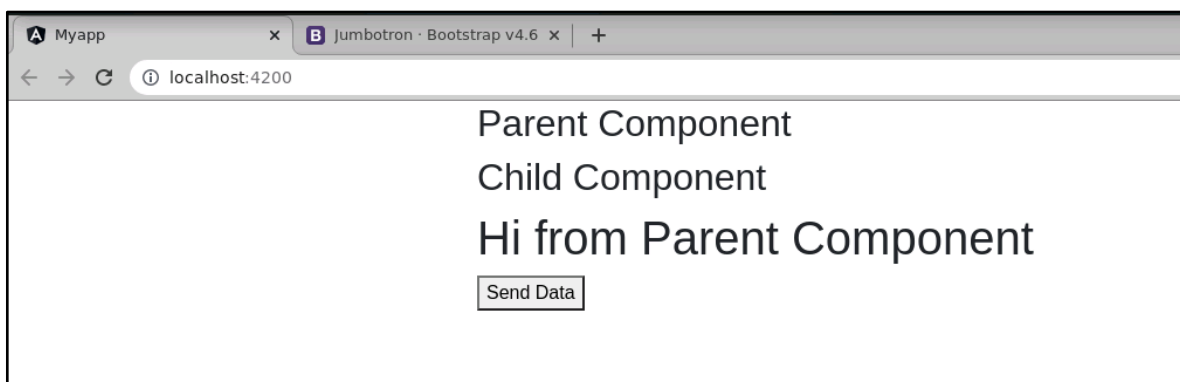
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.

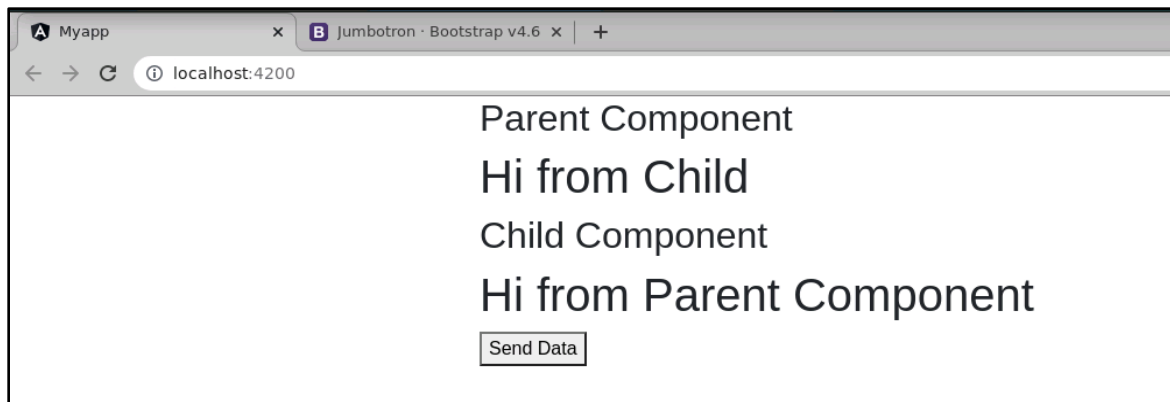
```

1.8 Open the browser and verify the output

1.9 **http://localhost: 4200/**



1.10 Click on the **Send Data** button to view the output shown below:



Note: The objective to set up communication between the nested components is achieved.