1. Wire up the click event handler

2. To raise the custom event, use the emit() method of the EventEmitter

3. PArent component bind to the custom event raised by the child component.

display-employee.component.html

```
<div class="panel panel-primary" (click)="handleClick()">
```

display-employee.component.ts

```
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
import {Employee} from '../models/employee.models';

@Component({
  selector: 'app-display-employee',
  templateUrl: './display-employee.component.html',
  styleUrls: ['./display-employee.component.css']
})
export class DisplayEmployeeComponent implements OnInit {
  @Input() employee: Employee;
  @Output() notify: EventEmitter<Employee> = new EventEmitter<Employee>();
constructor() { }

ngOnInit() {
  }
  handleClick() {
  this.notify.emit(this.employee);
```

list-employees.component.html

```
<h1 *ngIf="dataFromChild">
{{ dataFromChild?.name + ' ' + dataFromChild?.gender}}
</h1>
<div *ngFor="let employee of employees">
<app-display-employee [employee]="employee" (notify)="handleNofity($event)"></app-display-employee>

</div>
```

list-employees.component.html | list-employees.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Employee } from '../models/employee.models';
import { EmployeeService } from './employee.service';



@Component({
  templateUrl: './list-employees.component.html',
  styleUrls: ['./list-employees.component.css']
})
export class ListEmployeesComponent implements OnInit {
  employees: Employee[] = [];
  dataFromChild: Employee;
  constructor(private newEmployeeService: EmployeeService) { }

  ngOnInit() {
    this.employees = this.newEmployeeService.getEmployees();
  }
  handleNofity(eventData: Employee) {
    this.dataFromChild = eventData;
  }
}
```