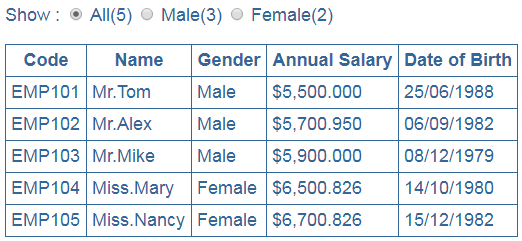
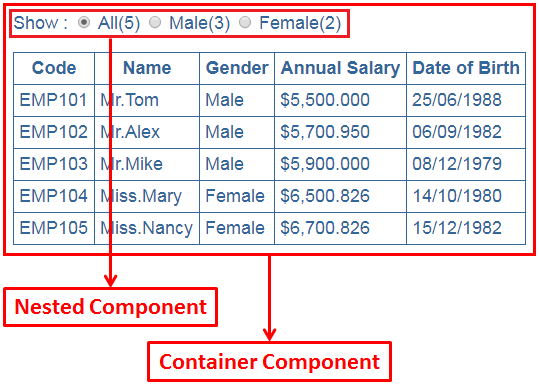
**Lesson18 container and nested components and use @Input to pass variable from parent Component to child component**

**Notes:-**

**1-we will want to use this Design to make filtration based on the values passed on the componentCountEmployee**



**What is the Parent Component and Child Component?**



**1-The Child Component is the component that inside the other component like the EmployeeCountComponent**

**2-The Parent Component is the component that holding the child components like the EmployeeListComponent**

|  |  |
| --- | --- |
| **Operator** | **Use to** |
| = | Assign a value |
| == | Compare two values |
| === | Compare two values and their types |

**Steps:-**

**1-we will create emp-count-component.html we write the following code**

<span class="radioClass">Show : </span>

<input type="radio" name="options" (click)="FilterEmployeeBasedOnGender('All')" ngModel/>

<span class="radioClass">{{"All(" + all + ")"}}</span>

<input name="options" type="radio" (click)="FilterEmployeeBasedOnGender('Male')" ngModel/>

<span class="radioClass">{{"Male(" + male + ")"}}</span>

<input name="options" type="radio" (click)="FilterEmployeeBasedOnGender('Female')" ngModel/>

<span class="radioClass">{{"Female(" + female + ")"}}</span>

**2-we apply the following code behind on the same component to make the variable input and be accessible from the container component to nested component**

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

import {Employee} from '../../Service/Employee';

import {EmployeeService} from '../../Service/employee.service';

import { DELEGATE\_CTOR } from '@angular/core/src/reflection/reflection\_capabilities';

@Component({

selector: 'app-employee-count',

templateUrl: './employee-count.component.html',

styleUrls: ['./employee-count.component.css']

})

export class EmployeeCountComponent implements OnInit {

//we make the @Input variable to be accessible to the parent component

@Input() all: number;

@Input() male: number;

@Input() female: number;

constructor(public employeeService:EmployeeService) {}

ngOnInit() {}

FilterEmployeeBasedOnGender(Gender:string){

this.employeeService.GetPersonBasedOnGender(Gender).subscribe(data =>

this.employeeService.Employees = data as Employee[]

);}}

**3-on the employee.component.html we write the following code**

//we Connect between the input variable and the methods that used to show the total number of employee based on its gender

<app-employee-count [all]="GetAllEmployees()" [male]="GetMaleEmployees()" [female]="GetFemaleEmployees()"></app-employee-count>

<table>

<thead>

<tr><th colspan="4">Employee Details</th></tr>

<tr>

<td></td>

<th style="text-align: left;">Full Name</th>

<th style="text-align: left;">Gender</th>

<th style="text-align: left;">Salary</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let item of employeeService.Employees;let count = index">

<td>{{count}}</td>

<td>{{GetFullName(item.Fname,item.Lname) |employeeTitle:item.Gender}}</td>

<td>{{item.Gender}}</td>

<td>{{item.Salary | currency:'USD':true:'1.4-4'}}</td>

</tr>

<tr \*ngIf="employeeService.Employees==0|| !employeeService.Employees">

<td colspan="4">

No Employees to display

</td>

</tr>

</tbody>

</table>

**4-on the employee-component.ts we write the following code**

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

import {Employee} from '../../Service/Employee';

import {EmployeeService} from '../../Service/employee.service';

import { DELEGATE\_CTOR } from '@angular/core/src/reflection/reflection\_capabilities';

@Component({

selector: 'app-employee',

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

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

})

export class EmployeeListComponent implements OnInit {

constructor(public employeeService:EmployeeService) {}

ngOnInit() {

this.ShowEmployees();}

GetFullName(Fname:string,Lname:string){

return Fname+Lname;}

ShowEmployees(){

this.employeeService.GetPersons().subscribe(data =>

this.employeeService.Employees = data as Employee[]

);}

//This is the Methods that Functions the map between the variable and the functions that the filtration the result

GetMaleEmployees():number{

return this.employeeService.Employees.filter(e => e.Gender === "Male").length;}

GetFemaleEmployees():number{

return this.employeeService.Employees.filter(e => e.Gender === "Female").length;}

GetAllEmployees():number{

return this.employeeService.Employees.length;}}