**Lesson27 Dependency Injection in Angular**

**Notes:-**

**1-Dependency Injection :-**  is responsible to creating and providing the instance to dependency (Service) the constructor section by registering the dependency (Service) on the app.module.ts level providers

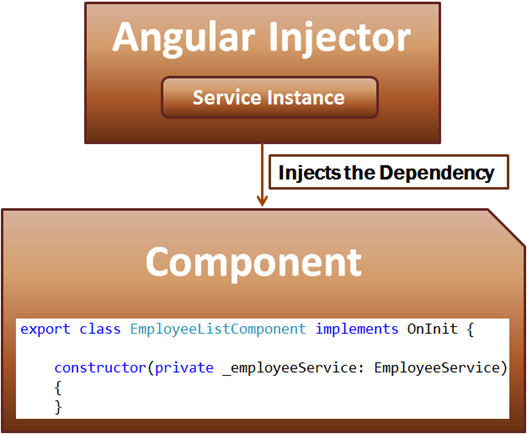
Notes:-

**1-We register a service with the angular injector by using the providers property of @Component decorator or @NgModule decorator.**

**2-When a component in Angular needs a service instance, it does not explicitly create it. Instead it just specifies it has a dependency on a service and needs an instance of it by including the service as a constructor parameter.**

**3-When an instance of the component is created, the angular injector creates an instance of the service class and provides it to component constructor.**

**4-So the component which is dependent on a service instance, receives the instance from an external source rather than creating it itself. This is called Dependency Injection.**



NgModule({

declarations: [

AppComponent,

EmployeeComponent,

HomeComponent,

MenuComponent,

PageNotFoundComponent,

EmployeeListComponent,

EmployeeDetailComponent],

imports: [

BrowserModule,

HttpClientModule,

RouterModule.forRoot(appRoutes)

],

//This is the part that we register the service on the app .module level

providers: [EmployeeService],

bootstrap: [AppComponent]

})

export class AppModule { }

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

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

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

@Component({

selector: 'app-employee-list',

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

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

})

export class EmployeeListComponent implements OnInit {

MessageStatus:string;

constructor(public employeeService:EmployeeService){

this.MessageStatus = "Loading Employees .... Please Wait";}

ngOnInit() {

debugger;

this.ShowEmployees();}

selectedEmployeeCountRadioButton: string = 'All';

onEmployeeCountRadioButtonChange(selectedRadioButtonValue: string): void {

debugger;

this.selectedEmployeeCountRadioButton = selectedRadioButtonValue;

}

GetFullName(Fname:string,Lname:string):string

{

return Fname+" "+Lname;

}

ShowEmployees()

{

debugger;

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

this.employeeService.Employees = data as Employee[],

error => {

this.MessageStatus =

'Problem with the service. Please try again after sometime';

// Notice here we are logging the error to the browser console

console.error(error);

});

}

}