**Component Communication:**

As we have to understand communication between two components using service, we will discuss about services before which covers our services topic as well as it will given clearer picture about component communication using services.

**Services in Angular:**

What is service in Angular?

1. Service is nothing but a typescript class which helps us to create reusable common shared functionalities between various modules and components. While discussing services, we have seen that how we can share same functionalities among multiple components with the help of demo.
2. Services can be injected into application using mechanism which is called as Dependency injection. Wherever we will have to use any service, we can inject that service in that respective service.
3. Services are an abstraction layer which consist of our application business logic.
4. Services are commonly used for making http requests to endpoints APIs to requests and receive the response.
5. Services can have methods, values which can share among multiple components.
6. **Example:** We have our Angular Application we have created. Here, assume we have default module App.
   1. **Module Name: App**
      1. **Components:**
         1. CustomerLogin
         2. CustomerRegistration
         3. CustomerFeedback
      2. **Services:**
         1. CheckInputService
         2. CommonDataService

So, example mentioned above where we can see App is our module which is having multiple components. But in services which we have. All services can used for all components.

1. Services are unlike components; we don’t have to include service in module.
2. We can create any number of services.
3. Services are used to share data between components.
4. We can inject services in components, this mechanism called as Dependency Injection.

**How to add/generate services in our application**

In order to add new service in application, we have to fire command.

**ng generate service <service\_name>**

If you want to create a service in any folder then we will have to add name of folder before service.

**ng generate service name\_of\_folder/<service\_name>**

**Step by Step Implementation of service for better understanding.**

**Step 1.** Let’s create a component first as Customer component.

**ng g c customer**

**Step 2.** Now, add new service as customer service.

**ng g s customerservice**

Now, open our service.ts file. This is code autogenerated by angular CLI. It has import injectable from @angular/core. There is also a decorator @Injectable. This decorator says this service is injectable in component.

@Injectable({

**provideIn: ‘root’**

})

The service is provided in “root” which means it’s available across application. It can be injected in any component. We can also change root and specify where we will have to provide it but by default its value is root means can inject anywhere.

**Step 3.** Now, we will add one method in our service.ts file.

getCustomerList(){

    const customers =[

      {custId:1,custName:'John'},

      {custId:2,custName:'James'},

      {custId:3,custName:'Kevin'},

      {custId:4,custName:'Kim'}

    ]

    return customers;

  }

**Step 4:** Now, we will have to use this service in customer component. Open customer.component.ts file.

1. import customerservice service into customer.component.ts.

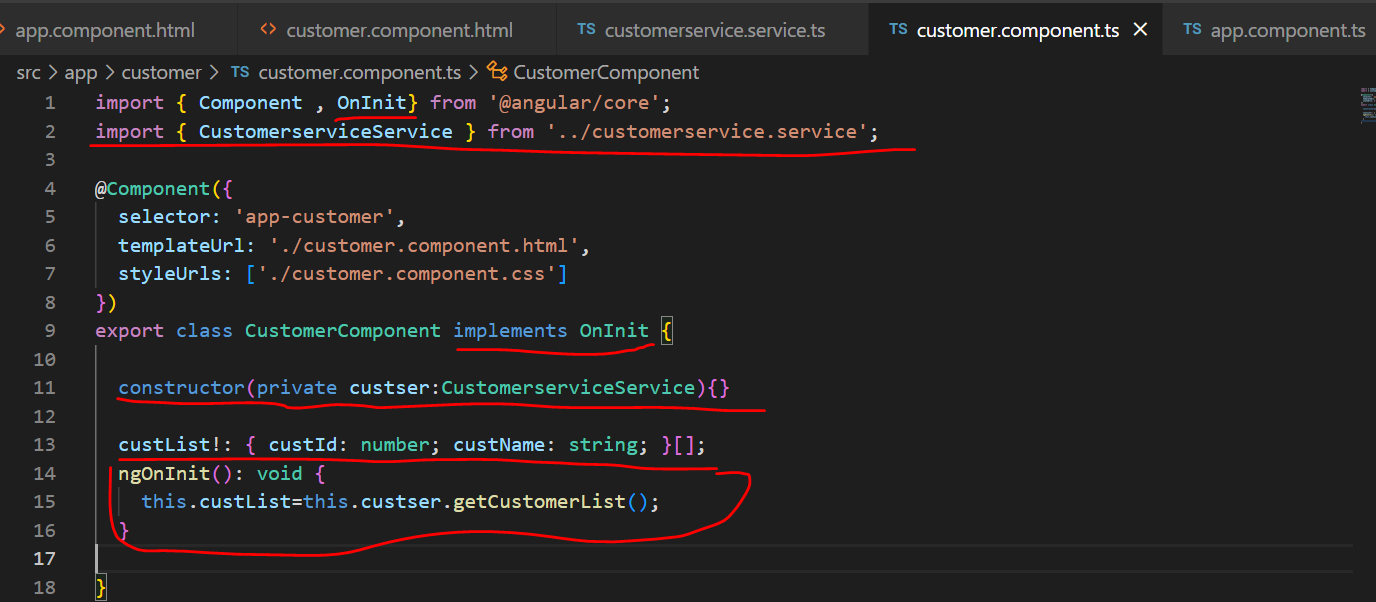
**import { CustomerserviceService } from './customerservice.service';**

1. inject it into constructor of class

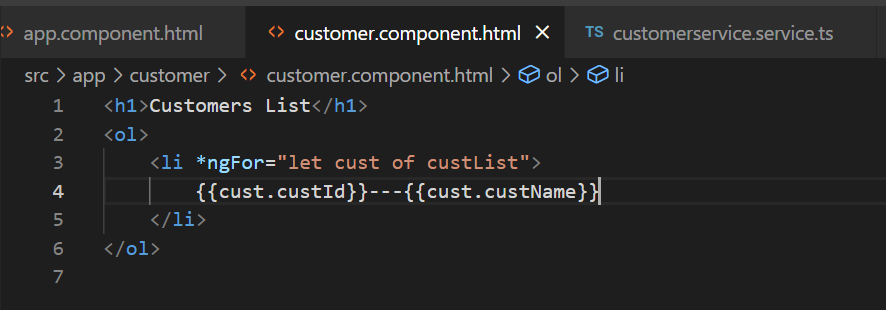
**constructor(private custservice:CustomerserviceService){}**

after this we will have to create fuction to get data from service.

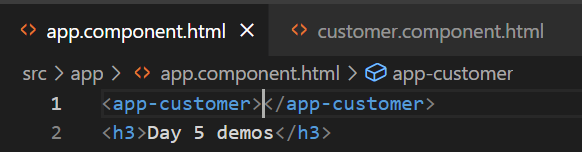
Finally our customerserviceservice.ts file will look like:



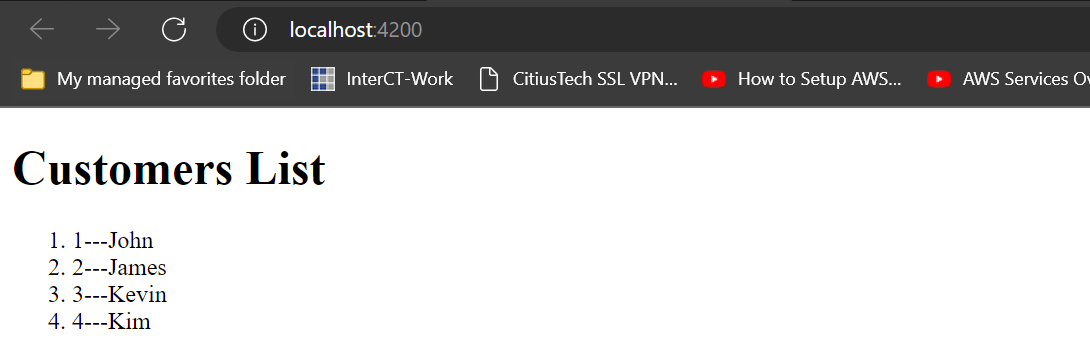
**Step 5.** Open customer.component.html file and add given below code:



**Step 6.** Now, add selector of customer component <app-customer> in app.component.html



After all, above steps our output should look like:



So, here we have seen how we can use services. Here our data is coming from customerservice. And in this way we can use this service in any number of components.

As we have seen that we can get values from service, but we can also send some values to service as well. This we will see when we will be working on using http with Angular.

So, one more thing that we have seen here is

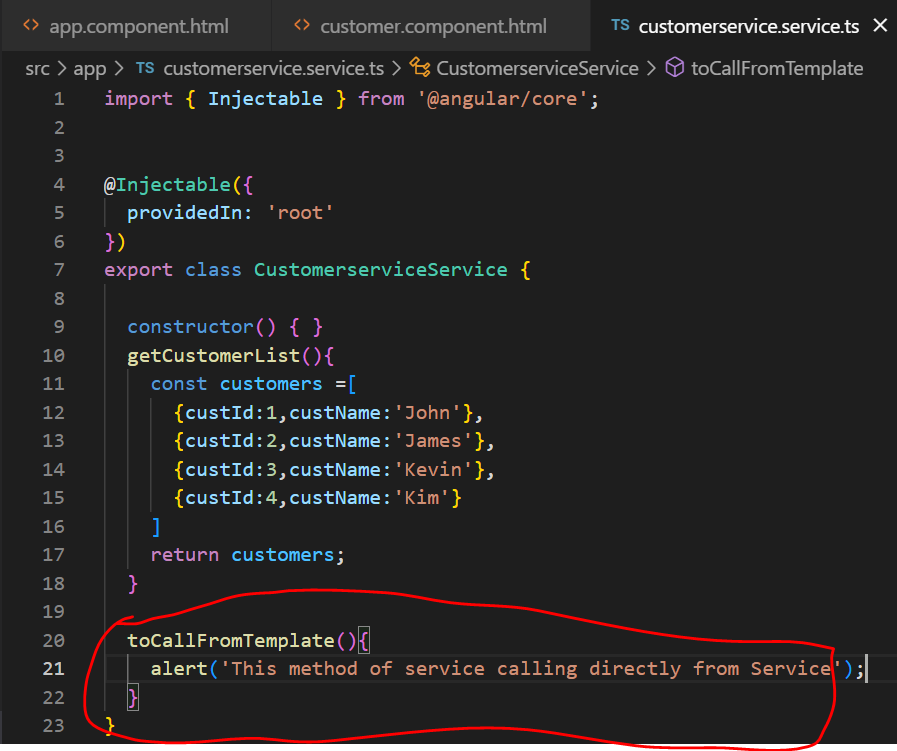
@Injectable

1. Dependency Injection(DI) which is one of design pattern.
2. Using DI we can provide data to components from injectable service class.
3. When angular CLI create service class for us, @injectable decorator will be added in this service class file. If we will create service class manually then we can add @Injectable.

There is one more way to access services directly into templates.

**How to do it?**

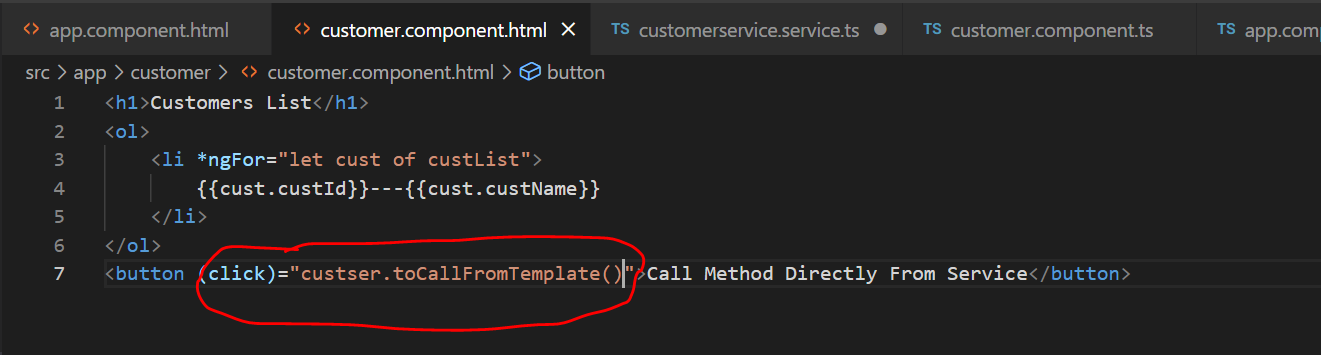
**Ans:** Let’s add one more method in customerservice.ts file.



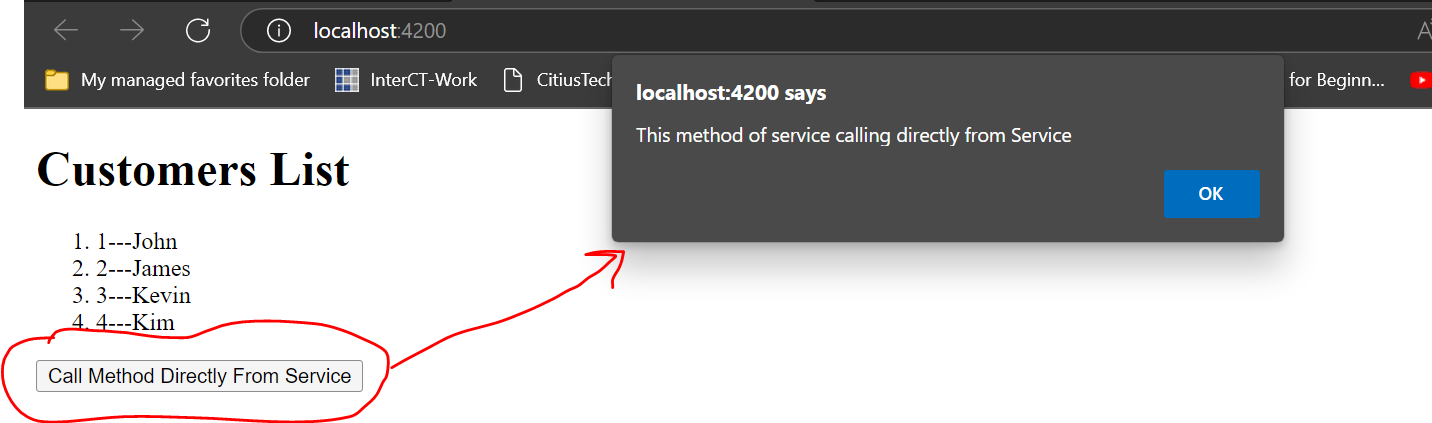
Now, go back to customer.component.ts and change access modifier in constructor from private to public.



Now, using this object custser, we can access method in services directly in customer template.



Output:



**Component Interaction using Service.**

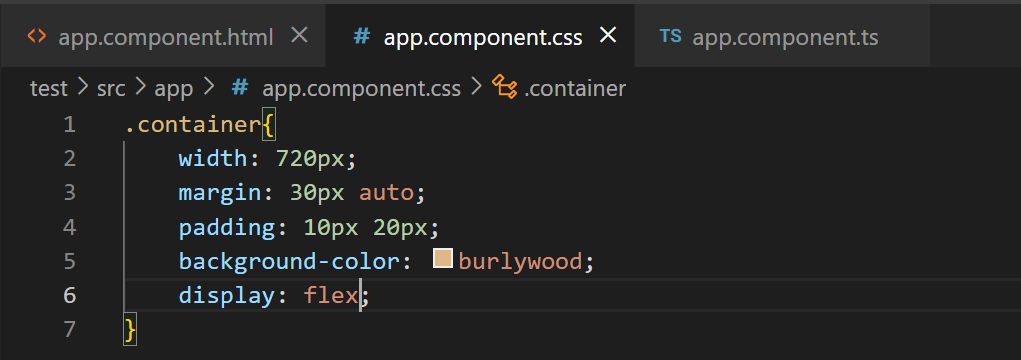
Component Interaction using Service is very easy.

For understanding this we will create two components. EmployeeInfo and DisplayEmployeeDetails. These two components are completely different. In EmployeeInfo component we will display List of Employee with one button for each employee to see their details and when we click on this button it should display Employee details in DisplayEmployeeDetails component. Let’s start implementing this requirement.

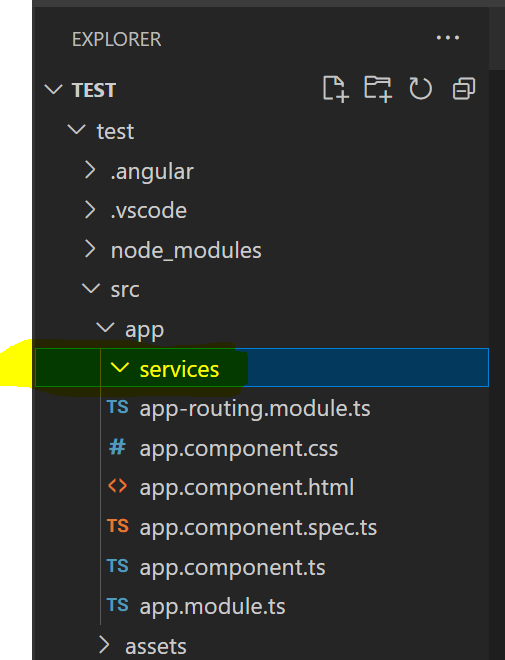
**Step 1.** Create a div in app.component.htmland provide class selector container as shown in given below picture.



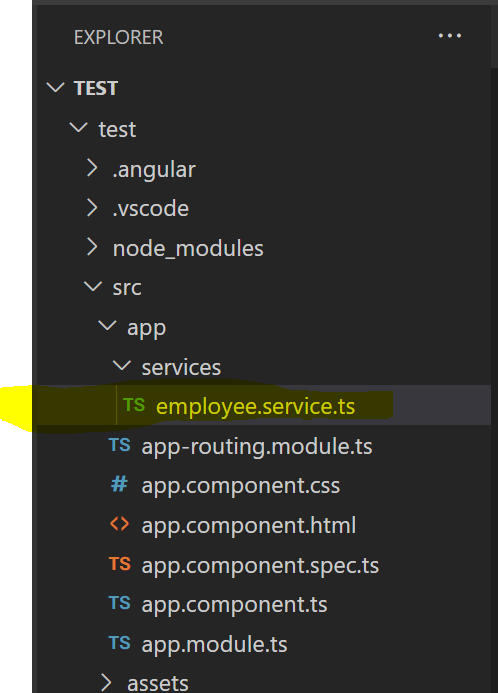
**Step 2.** Now in app.component.css file add some styling for this container.



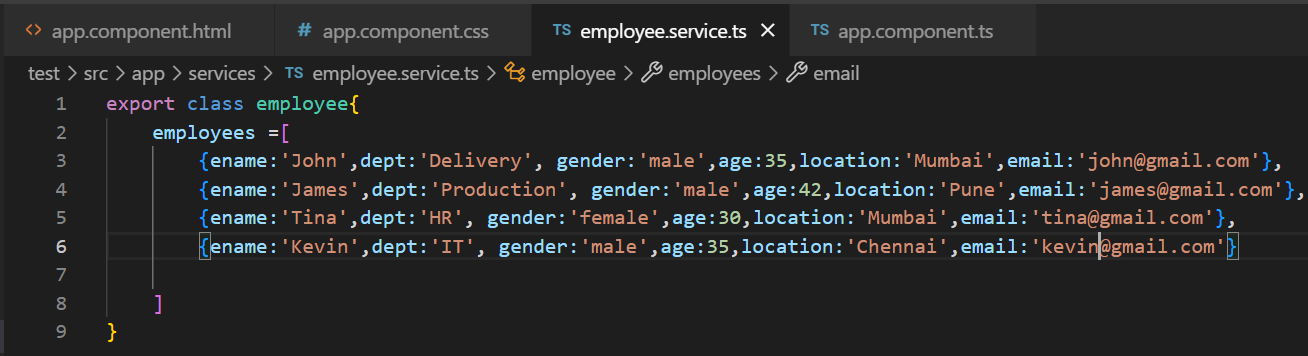
**Step 3.** Create a new folder as services in app folder.



**Step 4.** Inside this folder create new file as **employee.service.ts** file.



**Step 5.** Open employee.service.ts file and add given below code in it. We will create an array of employees in this file.



**Step 6:** Now, let’s inject this service in app component. At this point just try to understand that inject means we will have to get this service here in app component so, all it’s nested component can access this service. We will discuss dependency injection in depth when we’ll see Services.

For injecting service in app component we have to follow given below steps.

1. Import employee at top

import { employee } from './services/employee.service';

1. Create constructor of app.component.ts class to inject it.

constructor(private employees:employee){

  }

1. We will also have to specify providers too in @Component decorator.

@Component({

  selector: 'app-root',

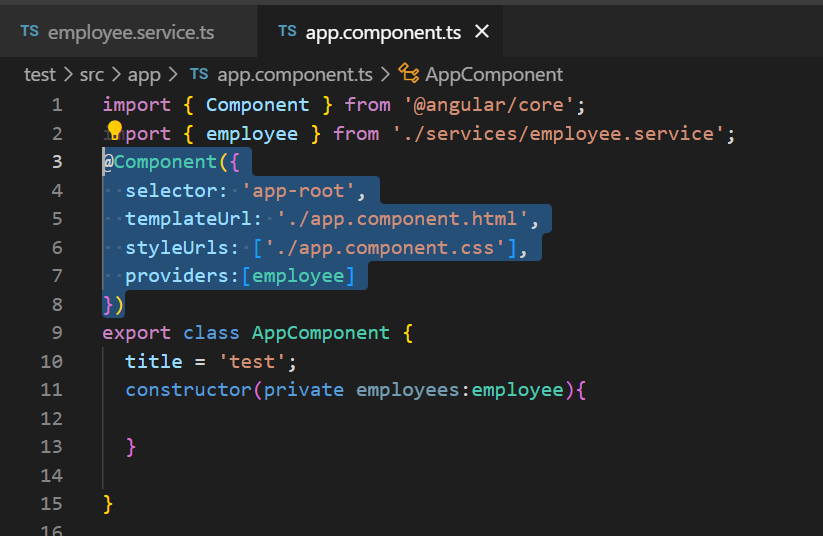
  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css'],

  providers:[employee]

})

Our final app.component.ts file will look like:



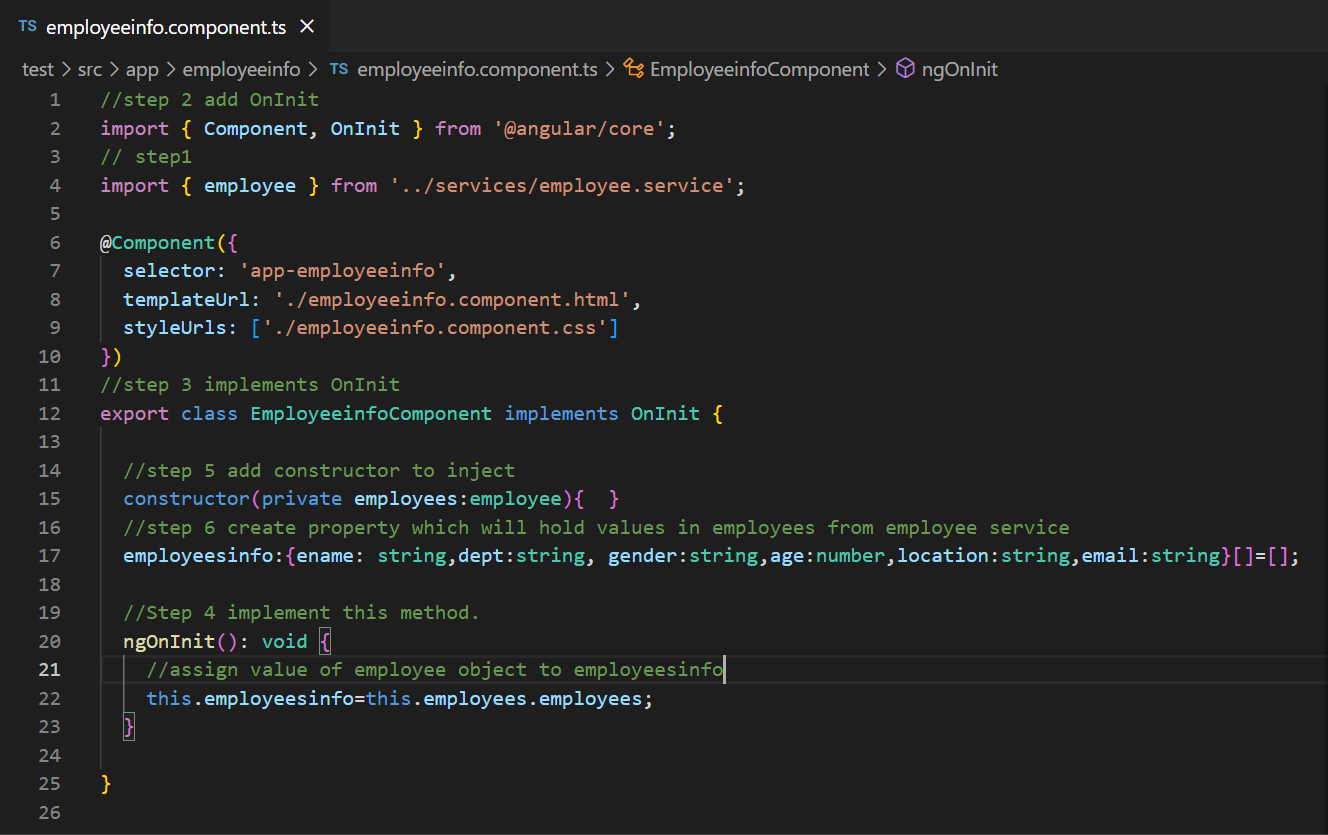
So, here we have injected employees service in app component and now, it can be use with child components of app components.

**Step 7.** Now, we will have to add new component employeeinfo.

And we have to get values of employees array from employeeservice.ts in employeeinfo.componen.ts file.

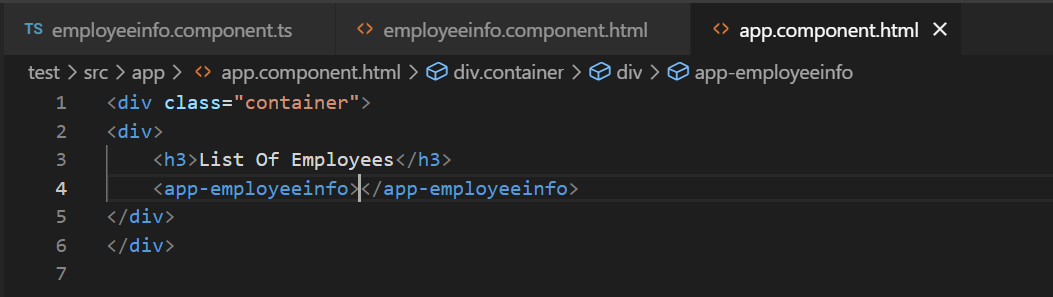
Final code of **employeeinfo.componen.ts** will be like :

Employeeinfo.component.ts

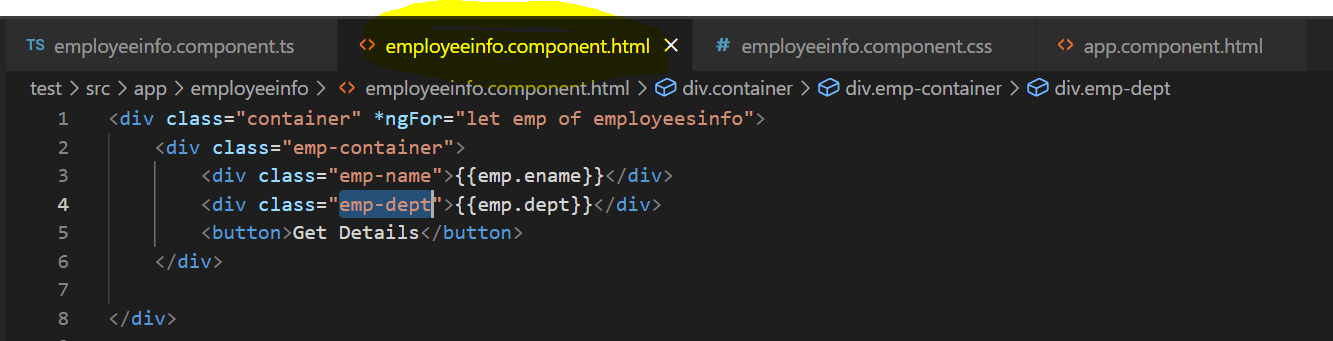


**Step 8.** Now, open app.component.html file and add selector of employeesinfo component.

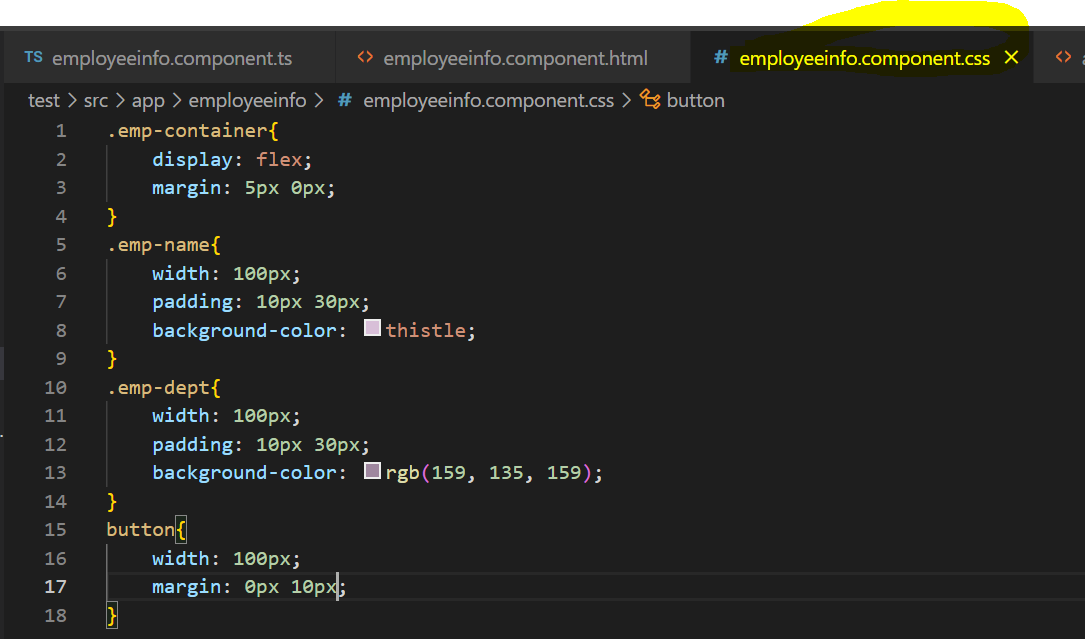
app.component.html file will look like:



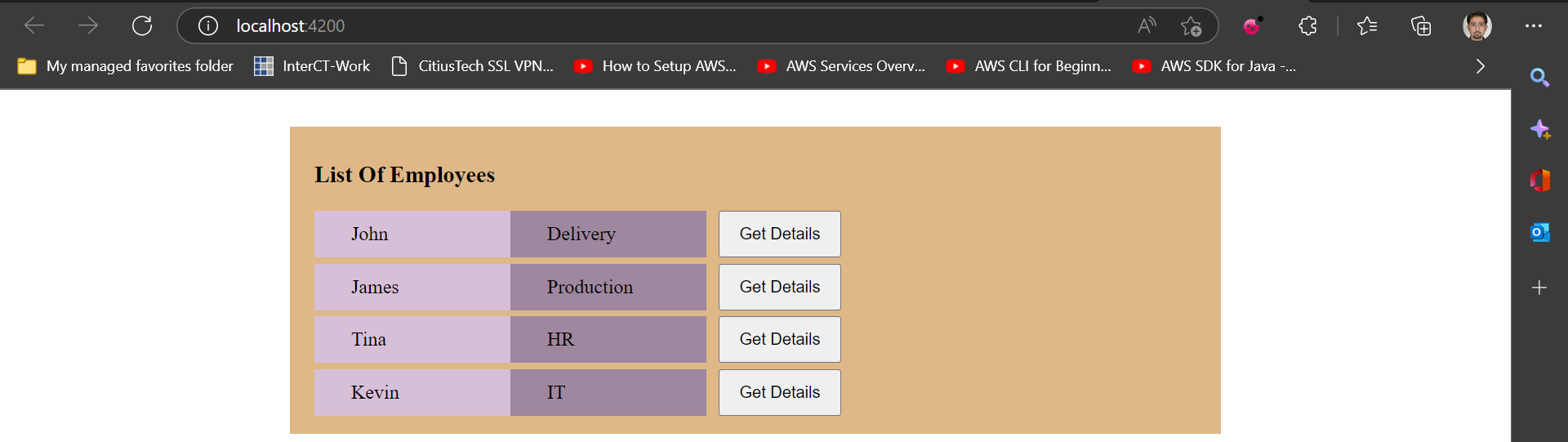
**Step 9.** Open employeeinfo.component.html file and add given below code.



Also add css in employeeinfo.component.css file.



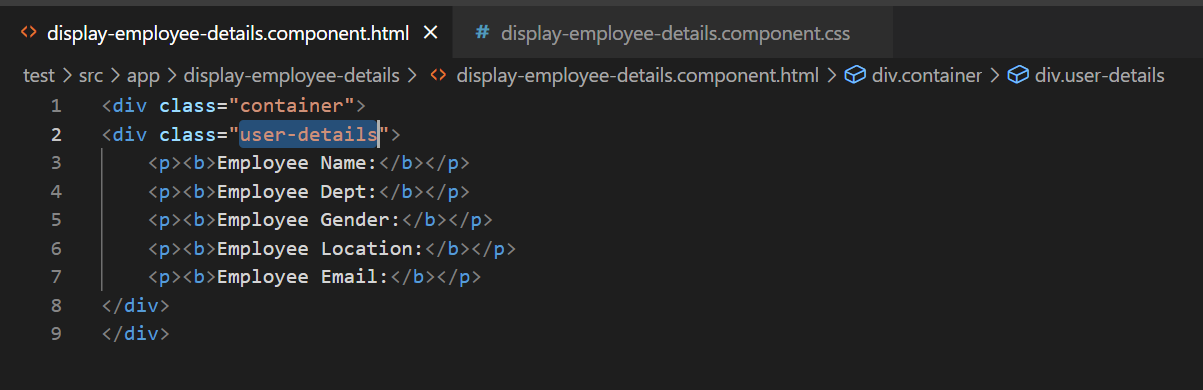
Till this point output of our application will look like:



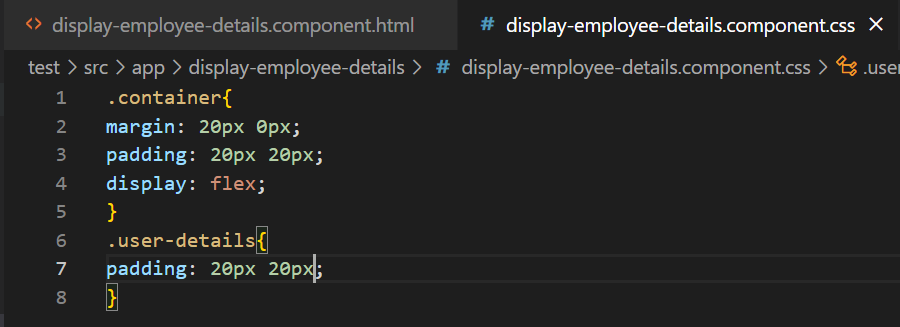
Now, our target is to get details of Employee on click event of button in another component.

**Step 10.** We will create new component as DisplayEmployeeDetails to display all details of employee.

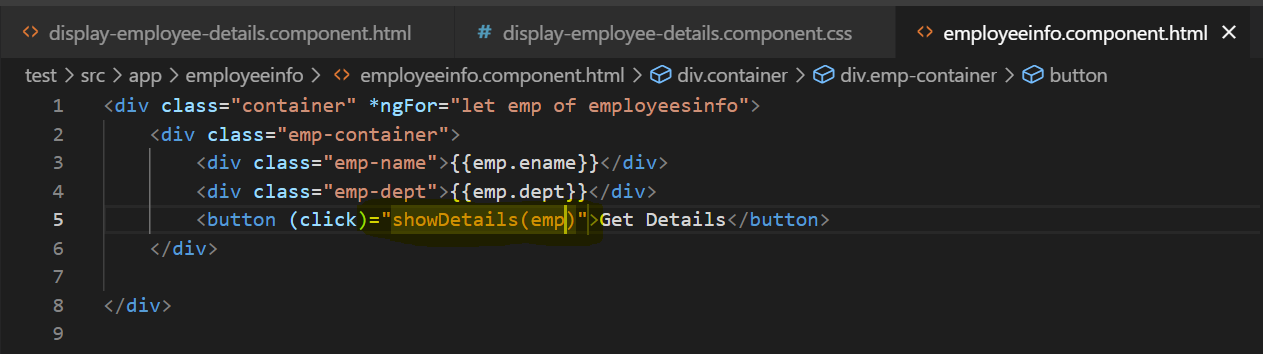
And open display-employee-details.component.html file first. Code in html file will look like.



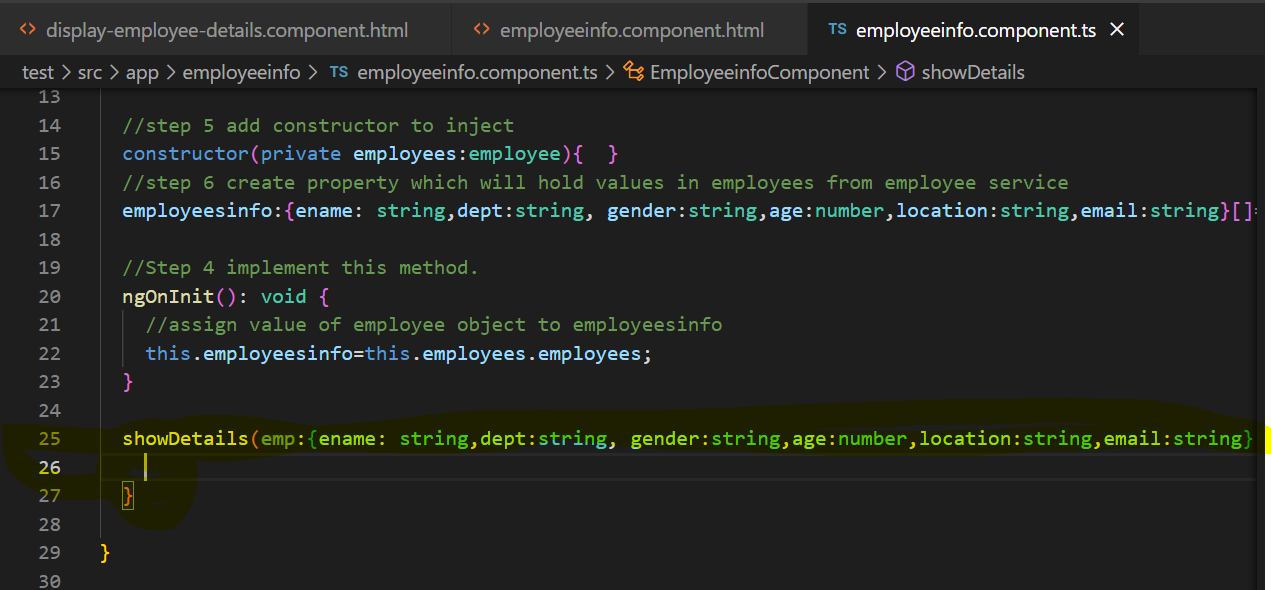
Add css in **display-employee-details.component.css.**



**Step 11.** Now, open employeeinfo.component.html file and for button we will add (click) event.



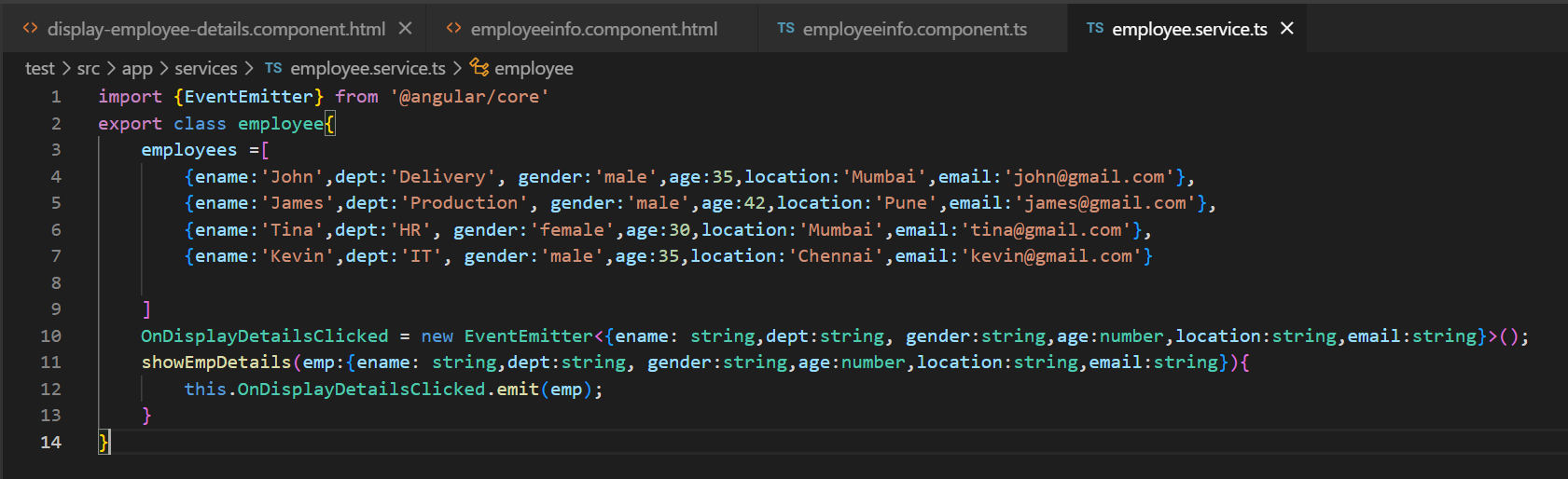
**Step 12.** Open employeeinfo.component.ts file and define showDetails() function.



But what we are supposed to do in this method.

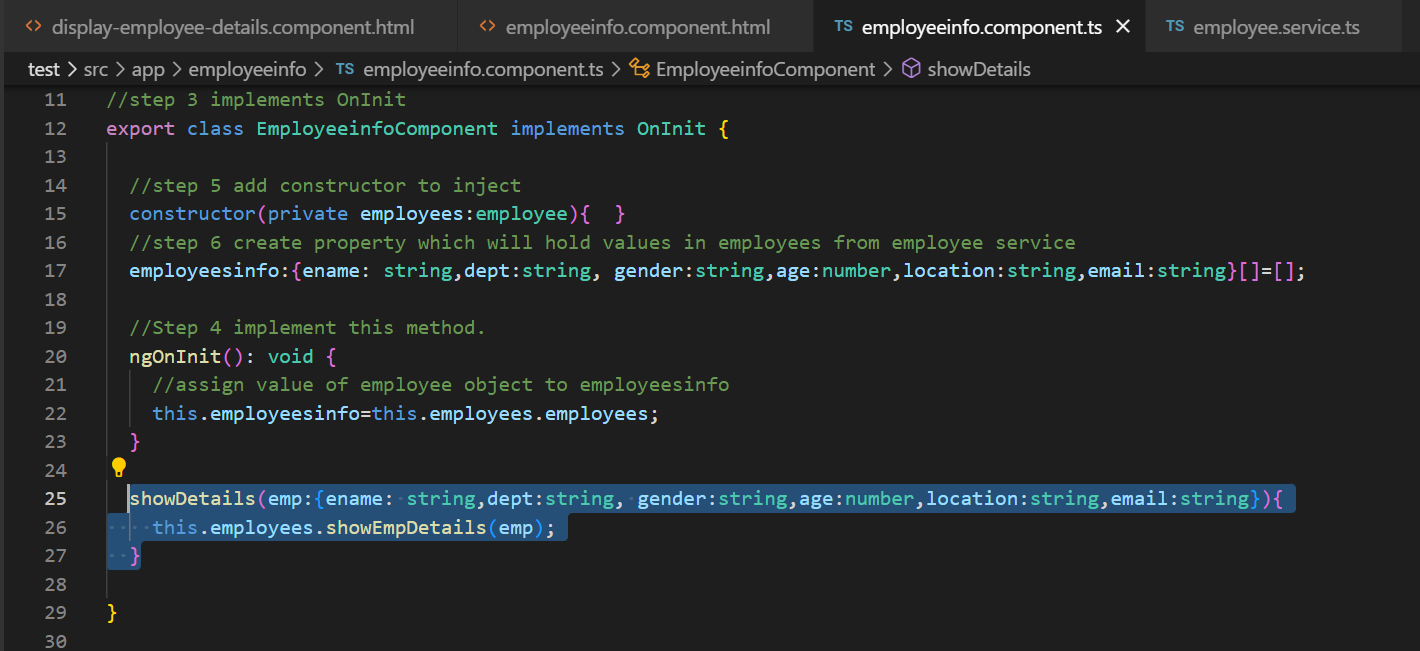
**Step 13.** Open our service.ts file and use EventEmmiter to emit data.

For this first we will have to import EventEmmiter in employees.service.ts file. And then create object to emit.



Now, we will have to call this method in showDetails() method in employeeinfo.component.ts file.

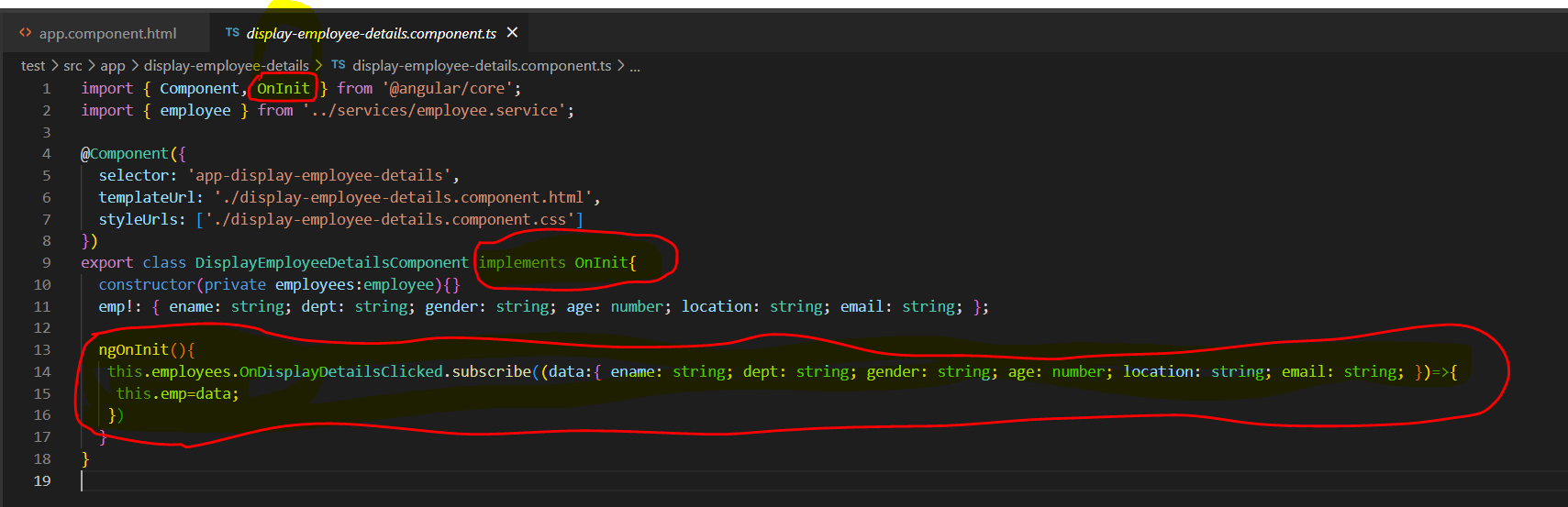
**Step 14.** Open employeeinfo.component.ts file and add given below code.



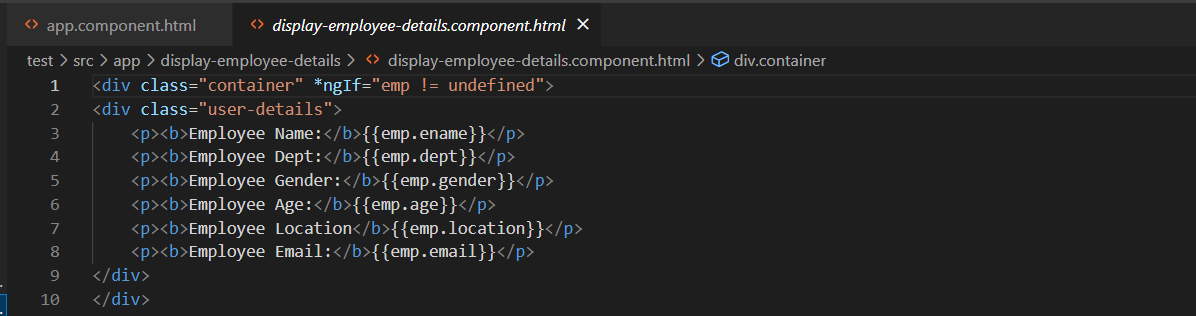
So, whenever we clicked on this button this method will be called and from this showDetails method, showEmpDetails() method in service will be called. When showEmpDetails() will be called , it will raise an event that is OnDisplayDetailClick event and when event will raise it will emit data.

Our emitted data will have to inject in display-employee-details.component.ts file.

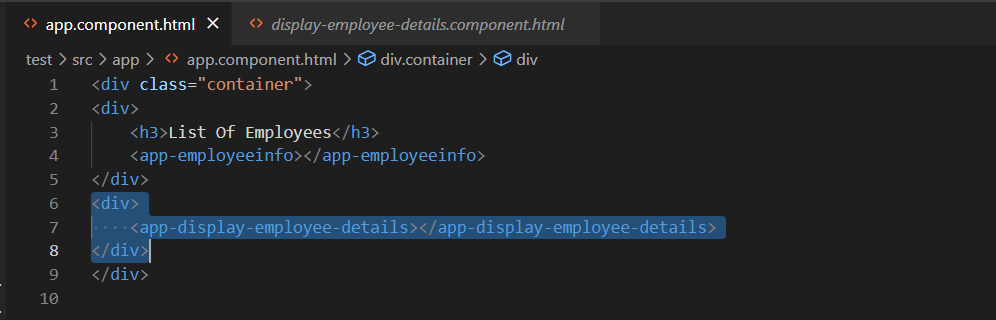
**Step 15:** Add given below code in **display-employee-details.component.ts** file



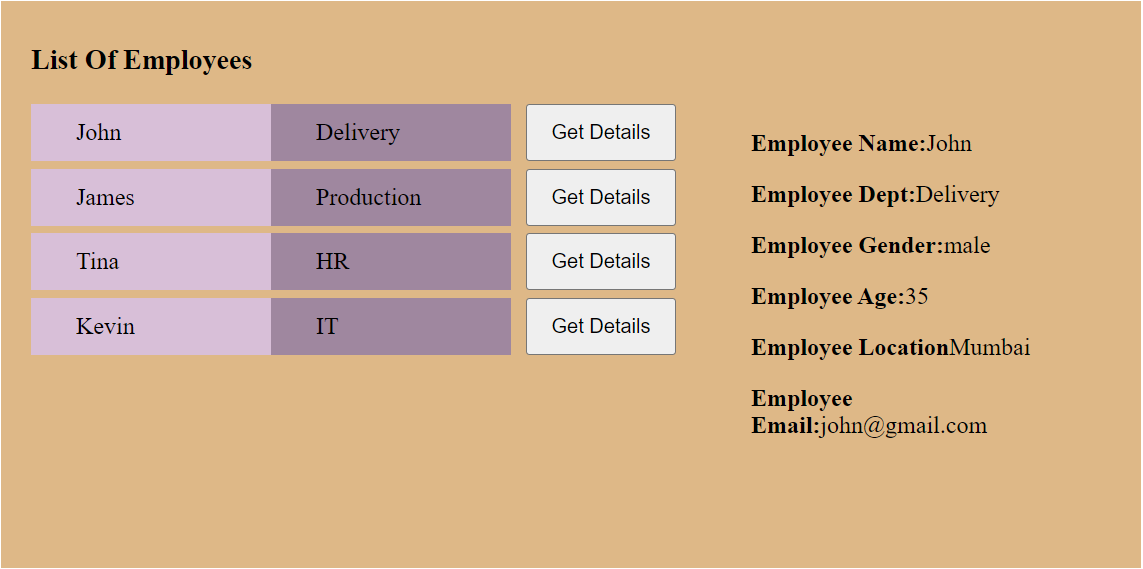
**Step 16.** Open display-employee-details.component.html file and add given below code.



**Step 17.** Open app.component.html file and add div to add selector of display-employee-details component.



After all above steps we will get below output.



[Angular 9 Tutorial For Beginners #58- Services - YouTube](https://www.youtube.com/watch?v=cznobRuS0UQ)

Directives:

[Directives in Angular Applications - YouTube](https://www.youtube.com/watch?v=LtT01ZCHRjk)

https://www.youtube.com/watch?v=LtT01ZCHRjk

Forms:

[Building Forms in Angular Apps | Mosh - YouTube](https://www.youtube.com/watch?v=hAaoPOx_oIw&list=RDCMUCWv7vMbMWH4-V0ZXdmDpPBA&start_radio=1&rv=hAaoPOx_oIw&t=294) <https://www.youtube.com/watch?v=hAaoPOx_oIw&list=RDCMUCWv7vMbMWH4-V0ZXdmDpPBA&start_radio=1&rv=hAaoPOx_oIw&t=294>

Full Course and pick required

[Introduction to Angular I Angular Basics | Angular 12+ - YouTube](https://www.youtube.com/watch?v=NMzl2pGOK_8&list=PL1BztTYDF-QNrtkvjkT6Wjc8es7QB4Gty)

<https://www.youtube.com/watch?v=NMzl2pGOK_8&list=PL1BztTYDF-QNrtkvjkT6Wjc8es7QB4Gty>