Services-Angular Notes

- A service is a class with a specific purpose.

- It is injected into other components and services that need it.

- Services are a great way to share information among classes that *don't know each other*.

- With services, we can access methods and properties across other components in the entire project.

- It must be declared as injectable using @Injectable decorator.

- When two different components need different instances of a service, it would be better to provide the service at the component level that needs the new and separate instance.

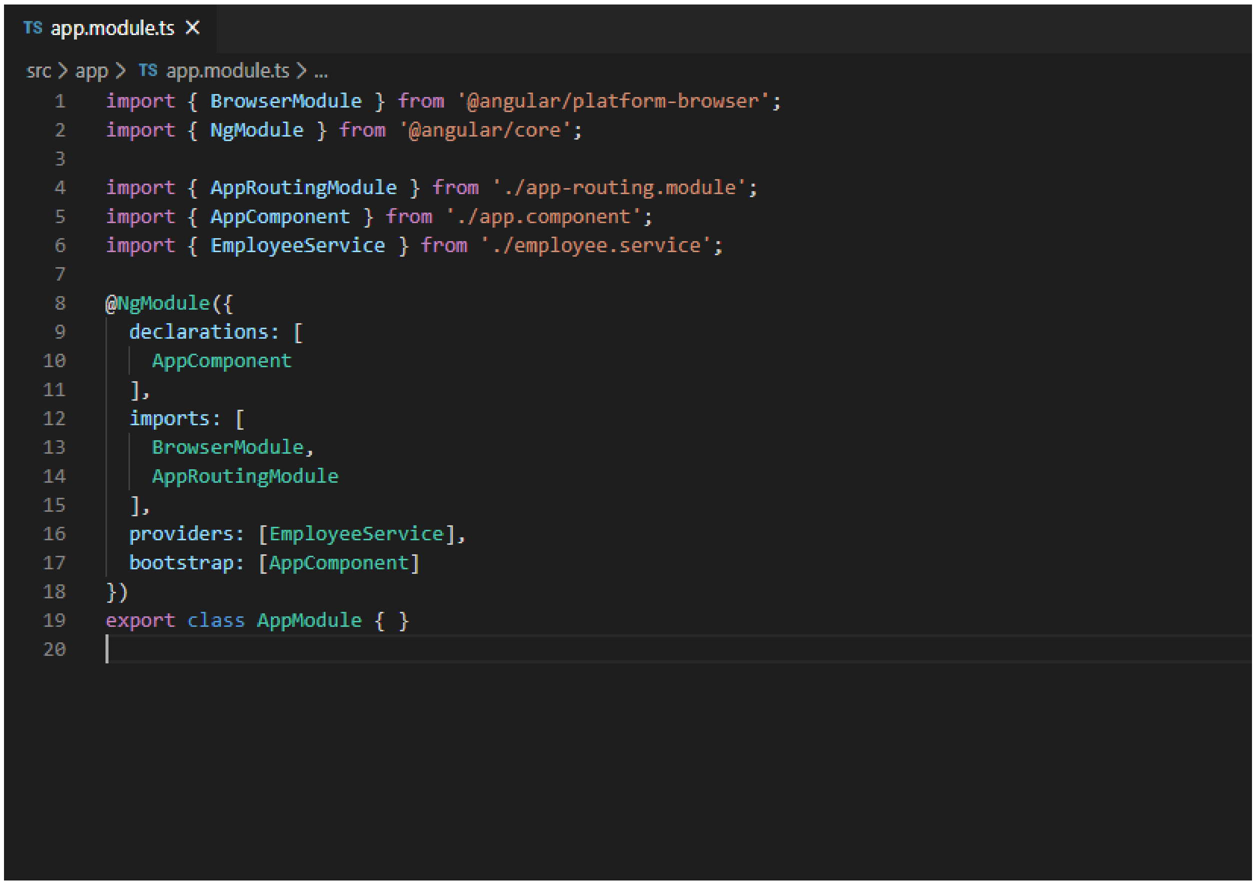
- A service is to be registered before it is used.
- A service provider is to be registered with the injector, or it won't know how to create the service.
- The providers property of @ Component or @ NgModule decorators is used to list services to be registered with the component or module.
- When a service is registered using module it is available to all components of that module.

Example 1

- Step 1: To create a service, we need to make use of the command line.

ng g service employee

- Step 2: Before creating a new service, we need to include the service created in the main parent **app.module.ts**.

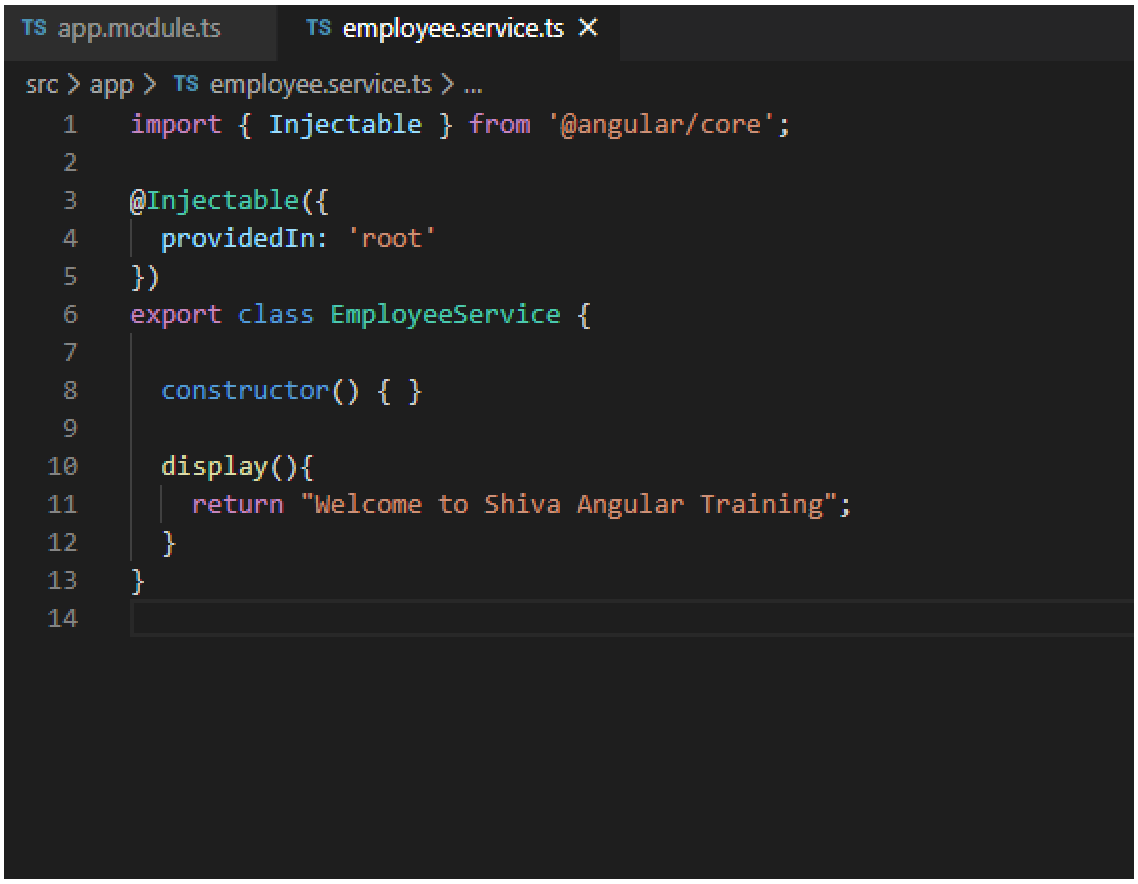


```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { EmployeeService } from './employee.service';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [EmployeeService],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

We have imported the Service with the class name and the same class is used in the providers.

- Step 3: Create a function in service class display() as below:-

```
display(){
  return "Welcome to Shiva Angular Training";
 }
```



```typescript
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class EmployeeService {

  constructor() { }

  display(){
    return "Welcome to Shiva Angular Training";
  }
}

```

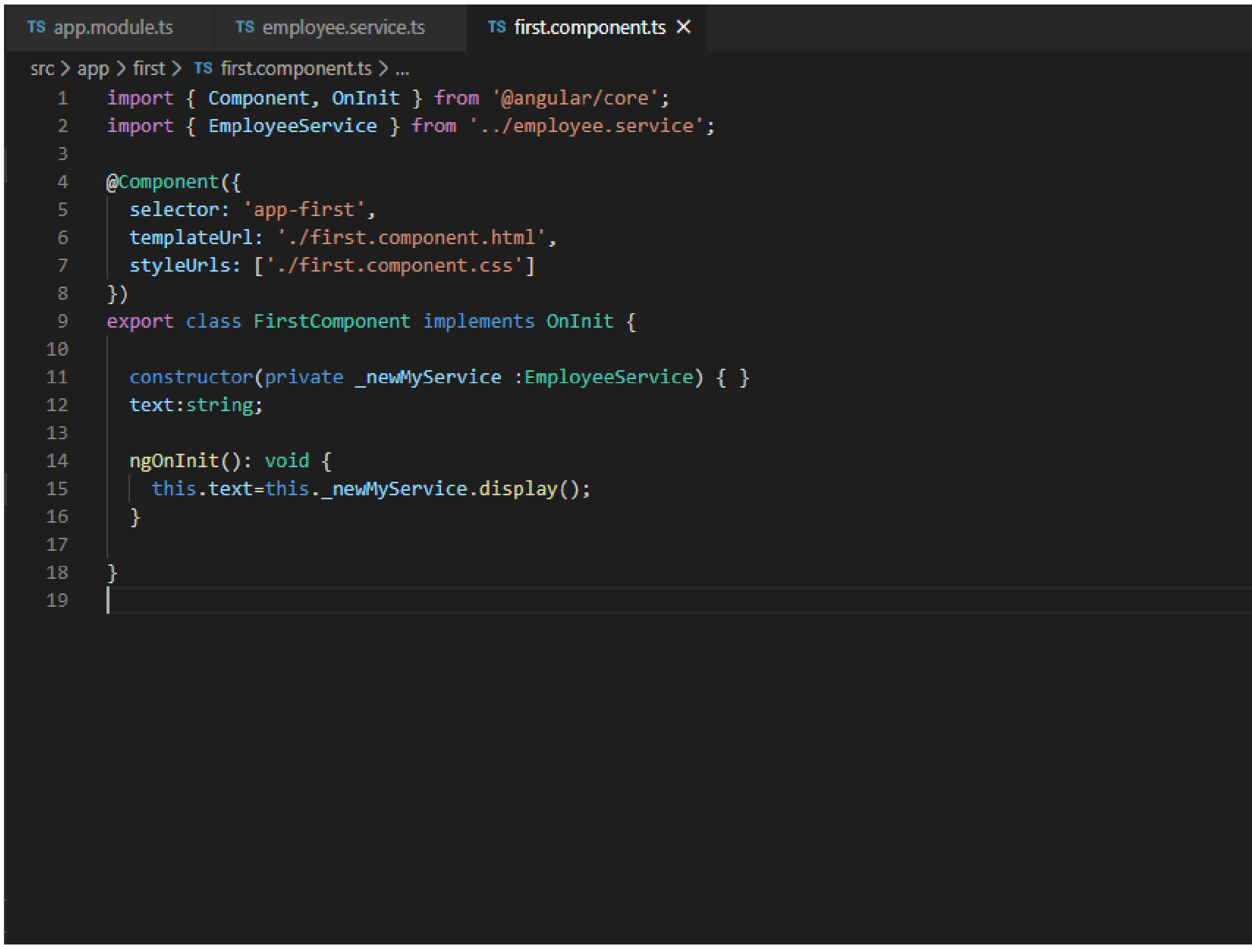
- Step 4 : Now create two components with name ‘first’ and ‘second’ in the project.

ng g component first

ng g component second

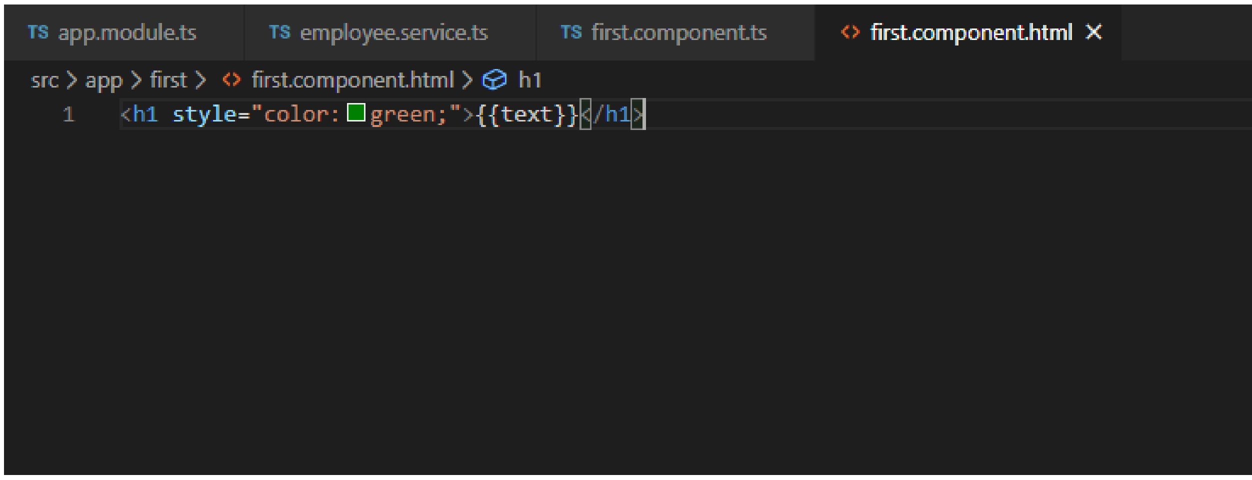
- Step 5: To use the service in the new component created. In component, inject service into a property by using a private parameter in constructor which is of service type. The **ngOnInit** function gets called by default in any component created. In the new component that we have created, we need to first import the service that we want and access the methods and properties of the same.

```
export class FirstComponent implements OnInit {

constructor(private _newMyService :EmployeeService) {
}
text:string;

ngOnInit(): void {
this.text=this._newMyService.display();
}

}
```



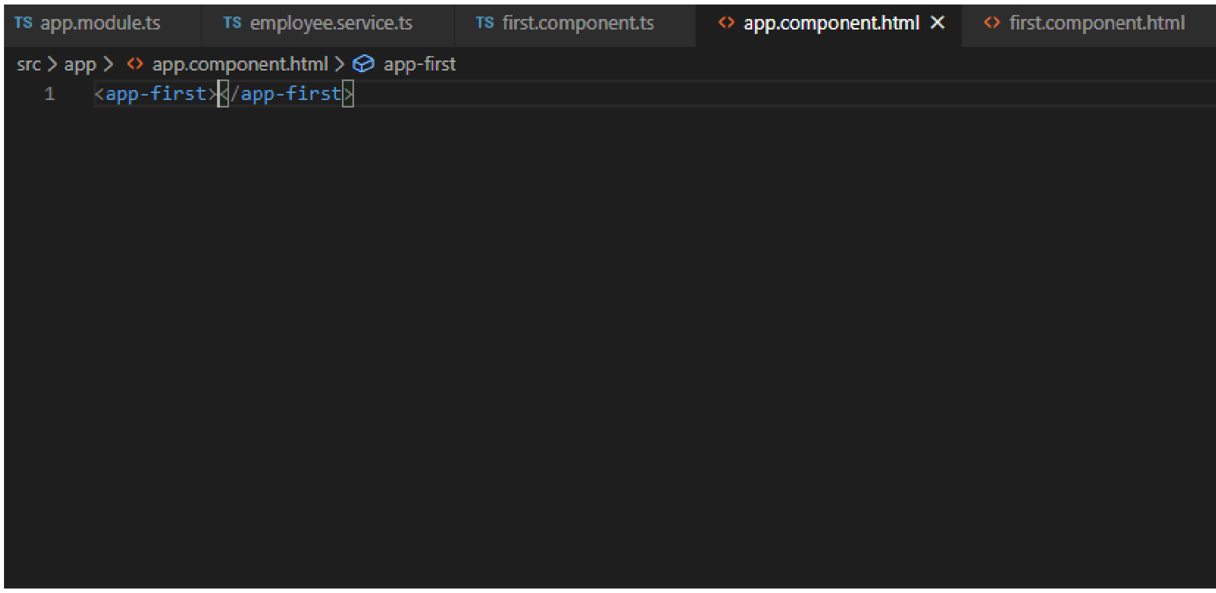
```
import { Component, OnInit } from '@angular/core';
import { EmployeeService } from '../employee.service';

@Component({
  selector: 'app-first',
  templateUrl: './first.component.html',
  styleUrls: ['./first.component.css']
})
export class FirstComponent implements OnInit {

  constructor(private _newMyService :EmployeeService) { }
  text:string;

  ngOnInit(): void {
    this.text=this._newMyService.display();
  }

}
```

➢ Step 6: The text is displayed in the component html as follows –



```
<h1 style="color:green;">{{text}}</h1>
```

- Step 7: The selector of the new component is used in the app.component.html file.

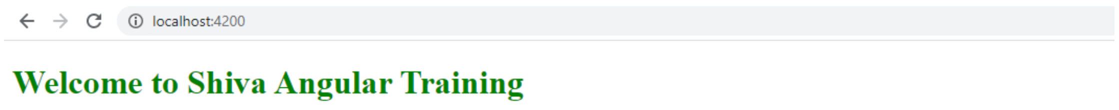


```
<app-first></app-first>
```

- Step 8: Next, navigate to your project's folder and run the local development server using the following command:

ng serve

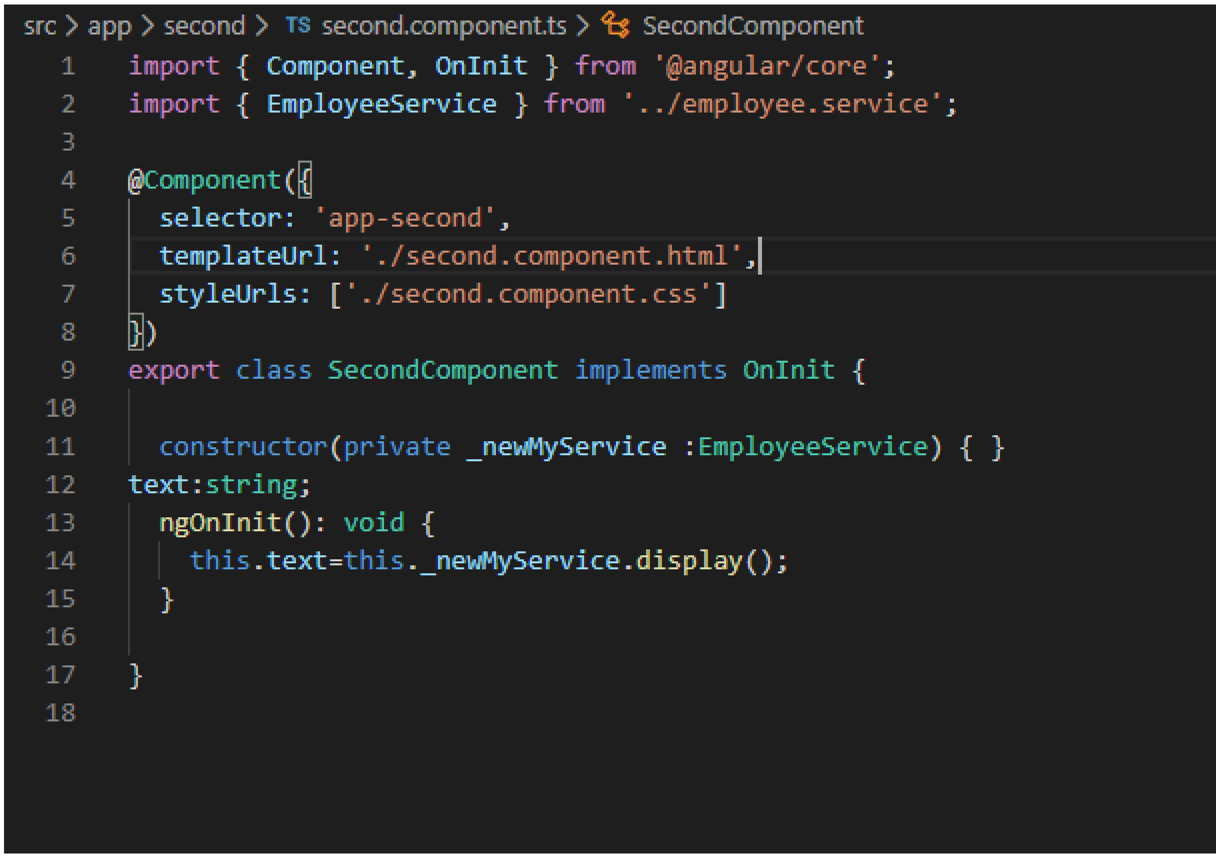
- A local development server will start listening on the http://localhost:4200/ address. Open your web browser and navigate to the http://localhost:4200/ address to see your app up and running. The contents from the above html file will be displayed in the browser as shown below –



localhost:4200

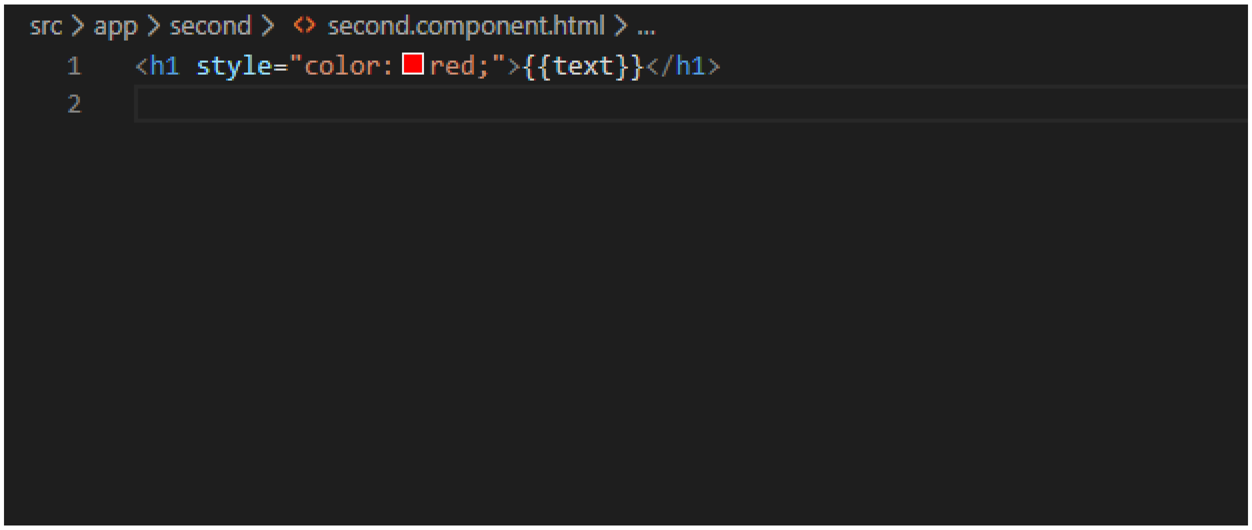
Welcome to Shiva Angular Training

➢ In the second component, the service can be accessed using the similar steps



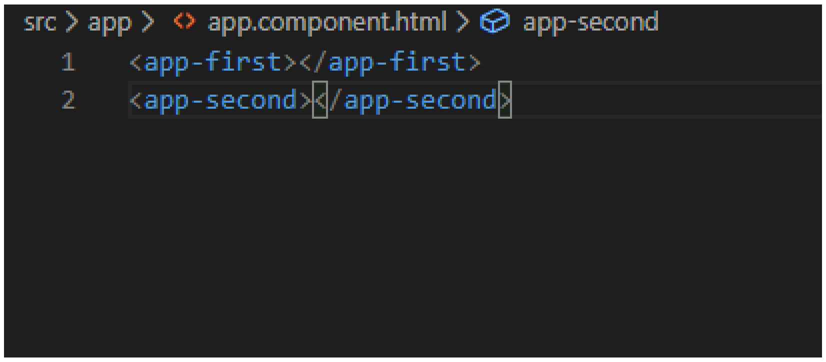
src > app > second > TS second.component.ts > SecondComponent

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

@Component({
  selector: 'app-second',
  templateUrl: './second.component.html',
  styleUrls: ['./second.component.css']
})
export class SecondComponent implements OnInit {

  constructor(private _newMyService :EmployeeService) { }
text:string;
  ngOnInit(): void {
    this.text=this._newMyService.display();
  }

}

```



src > app > second > second.component.html > ...

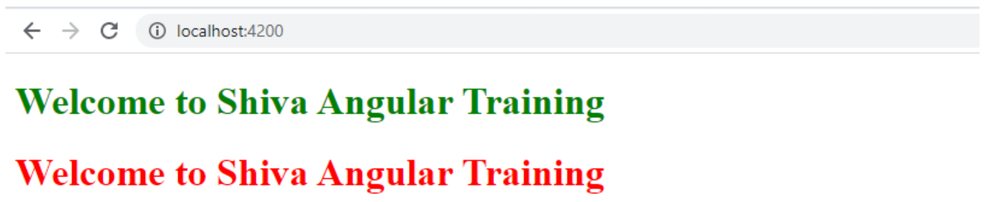
```html
<h1 style="color: red;">{{text}}</h1>
```



src > app > app.component.html > app-second

```html
<app-first></app-first>
<app-second></app-second>
```

- The contents from the above html file will be displayed in the browser as shown below –


localhost:4200

Welcome to Shiva Angular Training

Welcome to Shiva Angular Training

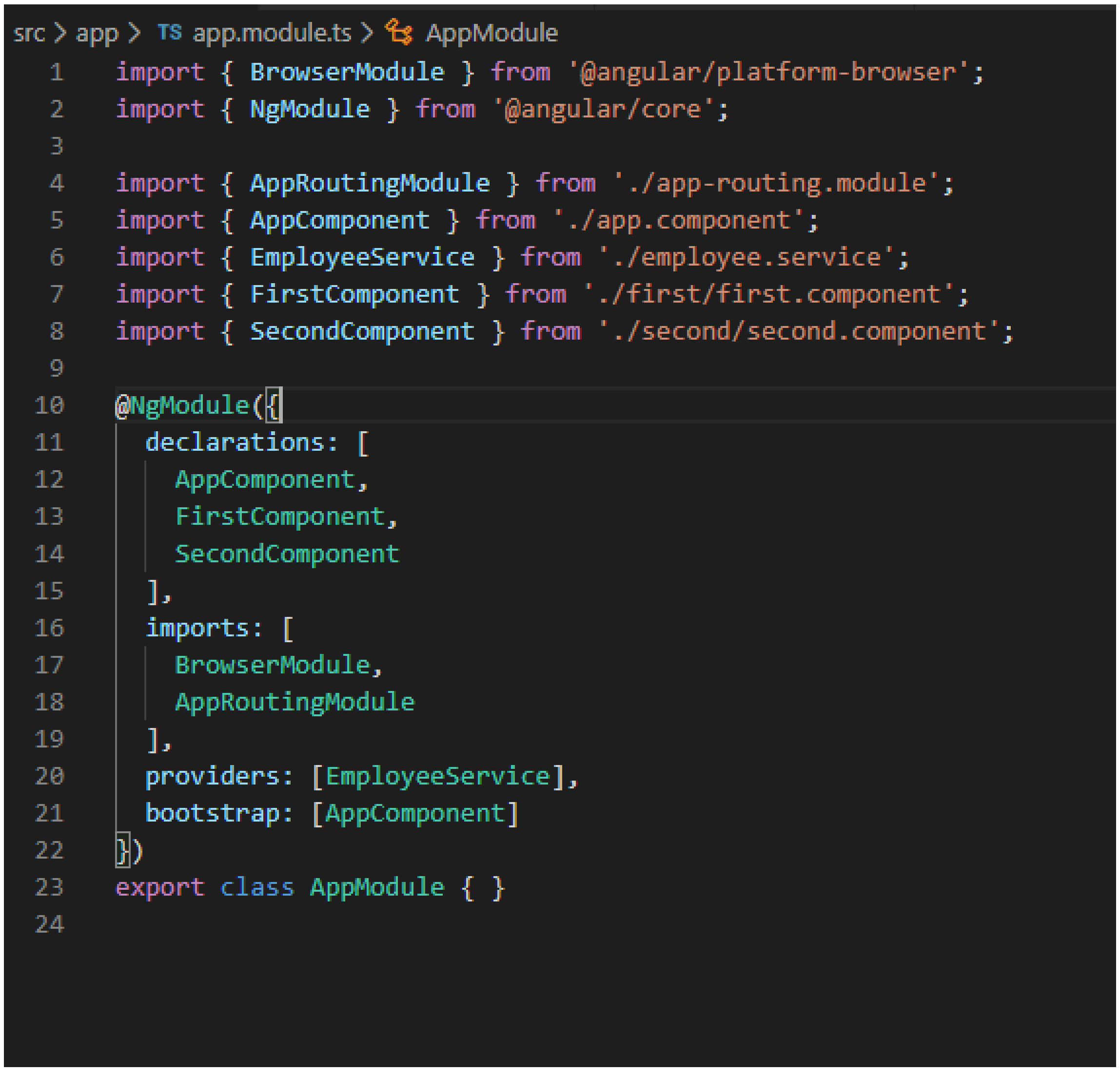
Example 2

Use the employee data specified as an array in the service class in the components.

- Step 1: To create a service, we need to make use of the command line.

ng g service employee

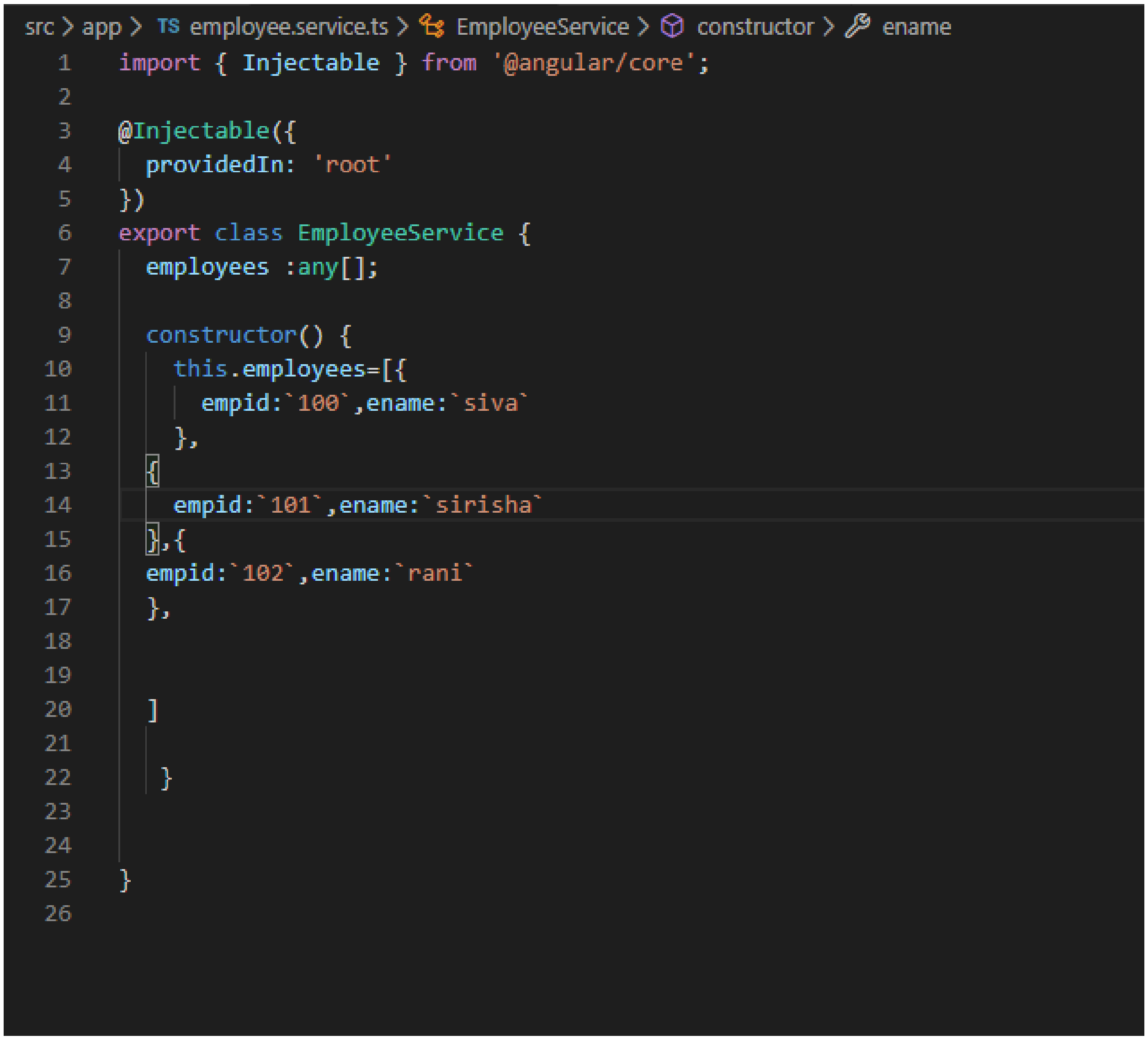
- Step 2: Before creating a new service, we need to include the service created in the main parent **app.module.ts**.


src > app > TS app.module.ts > AppModule

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { EmployeeService } from './employee.service';
import { FirstComponent } from './first/first.component';
import { SecondComponent } from './second/second.component';

@NgModule({
  declarations: [
    AppComponent,
    FirstComponent,
    SecondComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [EmployeeService],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

- Step 3: Create an array of employee with employee data in service class as below:-

```
export class EmployeeService {
  employees :any[];

  constructor() {
    this.employees=[{
      empid:`100`,ename:`siva`
    },
 {
   empid:`101`,ename:`sirisha`
 },{
 empid:`102`,ename:`rani`
 },
 ]
  }
}
```



src > app > TS employee.service.ts > EmployeeService > constructor > ename

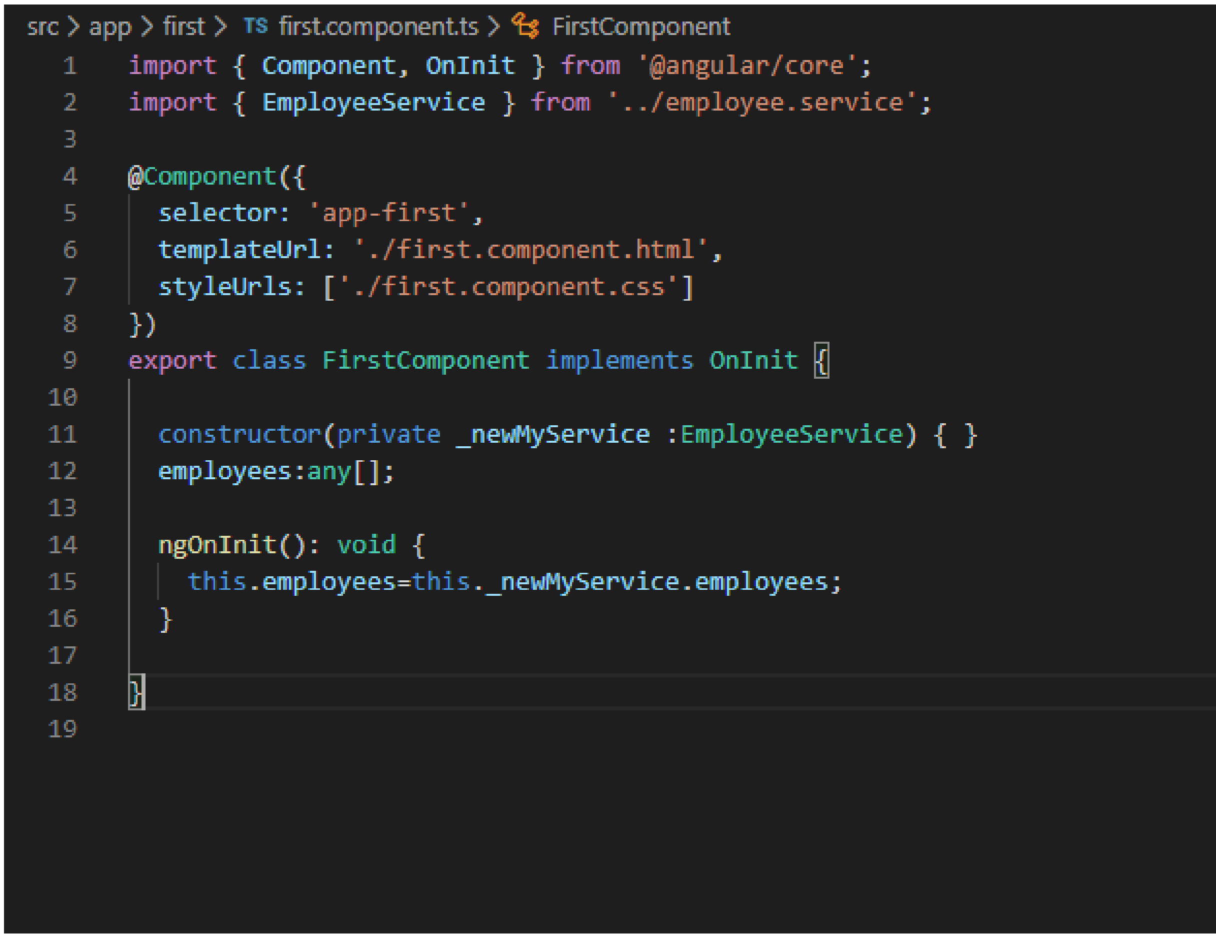
```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class EmployeeService {
  employees :any[];

  constructor() {
    this.employees=[{
      empid:`100`,ename:`siva`
    },
  {
    empid:`101`,ename:`sirisha`
  },{
  empid:`102`,ename:`rani`
  },


  ]

   }


}

```

- Step 4 : Now create two components with name ‘first’ and ‘second’ in the project.

```
ng g component first

ng g component second
```

- In the new component that we have created, we need to first import the service that we want and access the methods and properties of the same.

```
export class FirstComponent implements OnInit {

  constructor(private _newMyService :EmployeeService
) { }
  employees:any[];

  ngOnInit(): void {
    this.employees=this._newMyService.employees;
  }

}
```

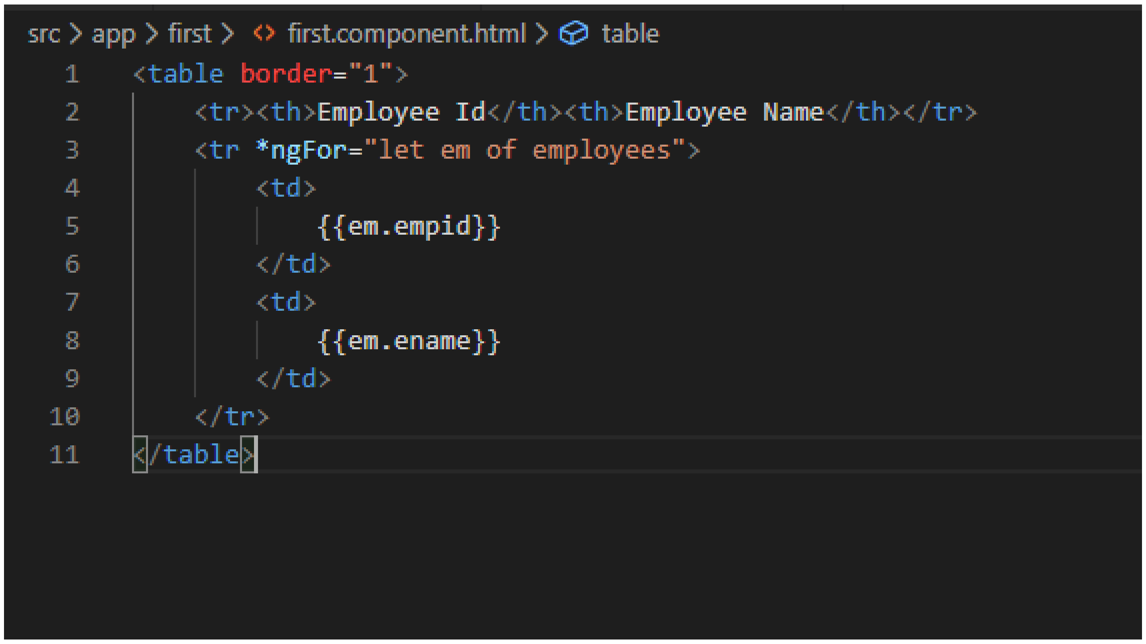

src > app > first > TS first.component.ts > FirstComponent

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

@Component({
  selector: 'app-first',
  templateUrl: './first.component.html',
  styleUrls: ['./first.component.css']
})
export class FirstComponent implements OnInit {

  constructor(private _newMyService :EmployeeService) { }
  employees:any[];

  ngOnInit(): void {
    this.employees=this._newMyService.employees;
  }

}

```

- Step 5: To display the data in firstcomponent.html

```
<table border="1">
   <tr><th>Employee Id</th><th>Employee Name</th></tr>
   <tr *ngFor="let em of employees">
      <td>
         {{em.empid}}
      </td>
      <td>
         {{em.ename}}
      </td>
```

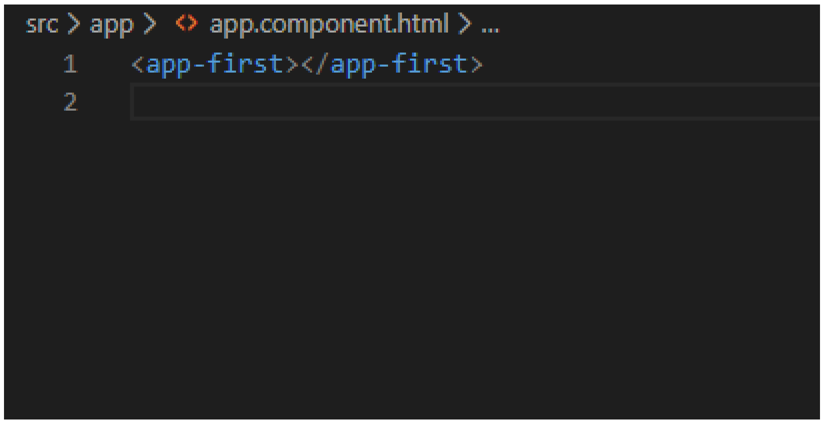
</tr>
</table>



src > app > first > first.component.html > table

```html
<table border="1">
    <tr><th>Employee Id</th><th>Employee Name</th></tr>
    <tr *ngFor="let em of employees">
        <td>
            {{em.empid}}
        </td>
        <td>
            {{em.ename}}
        </td>
    </tr>
</table>
```

- Step 7: The selector of the new component is used in the app.component.html file.



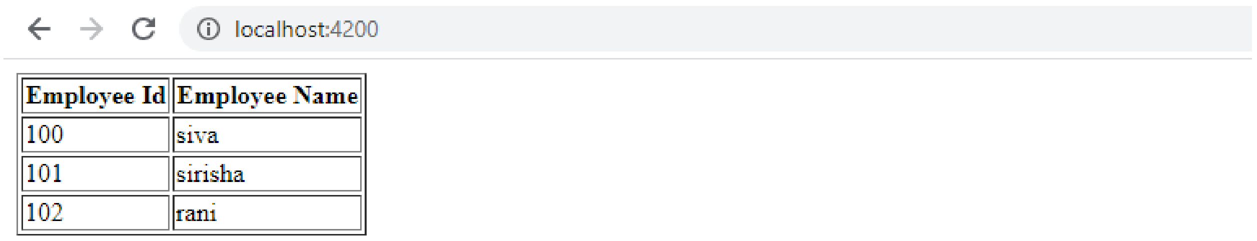
src > app > app.component.html > ...

```html
<app-first></app-first>

```

- Step 8: Next, navigate to your project's folder and run the local development server using the following command:

ng serve

- A local development server will start listening on the http://localhost:4200/ address. Open your web browser and navigate to the http://localhost:4200/ address to see your app up and running. The contents from the above html file will be displayed in the browser as shown below –



Employee Id	Employee Name
100	siva
101	sirisha
102	rani