Angular – Service Notes

Sharing of data is one of the requirements in angular application. Sometimes we want to use same data across components. So to do so you have 2 approaches

1) Create that data in each component individually - but there is drawback with that approach we are re-writing the code which is against the approach of DRY(DONT REPEAT YOURSELF) and code also become lengthy , complex and less testable. So next approach is Services

2) Service is a centralized location in angular application where you can keep your data which you want to share across components.

Any modifications done in service will automatically reflected back to all the components who is injecting this service because services are singleton (Single instance of that service will get created)

To create service use ng generate command: **ng g s services/userService**

All services class will use Injectable () decorator because we can inject service in any component

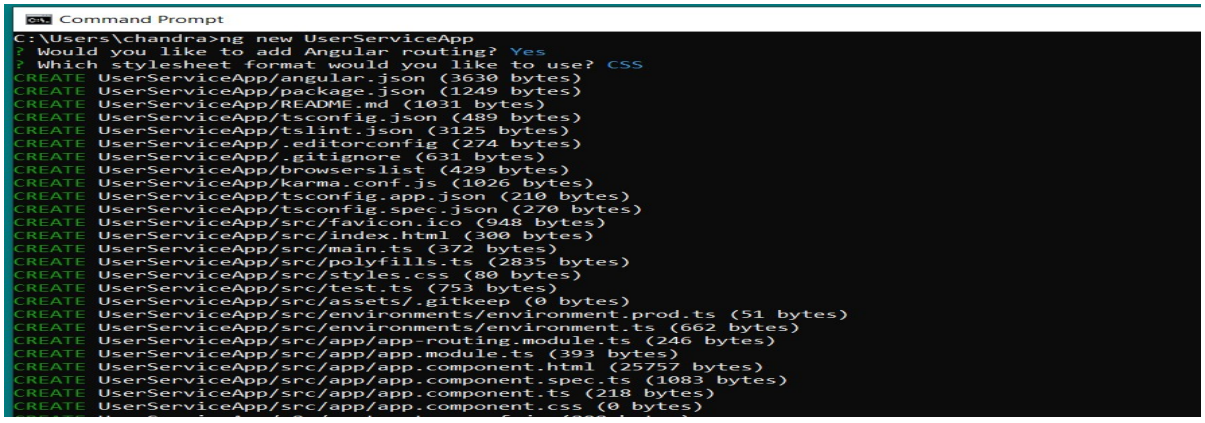
After that we need to register our service the way we have registered our component to app.module.ts in declarations. All service will registered in app.module.ts providers [] array.

After registration we need to use DI(Dependency Injection) to inject the reference of that service in any component .

After DI we can call all methods define in that service class

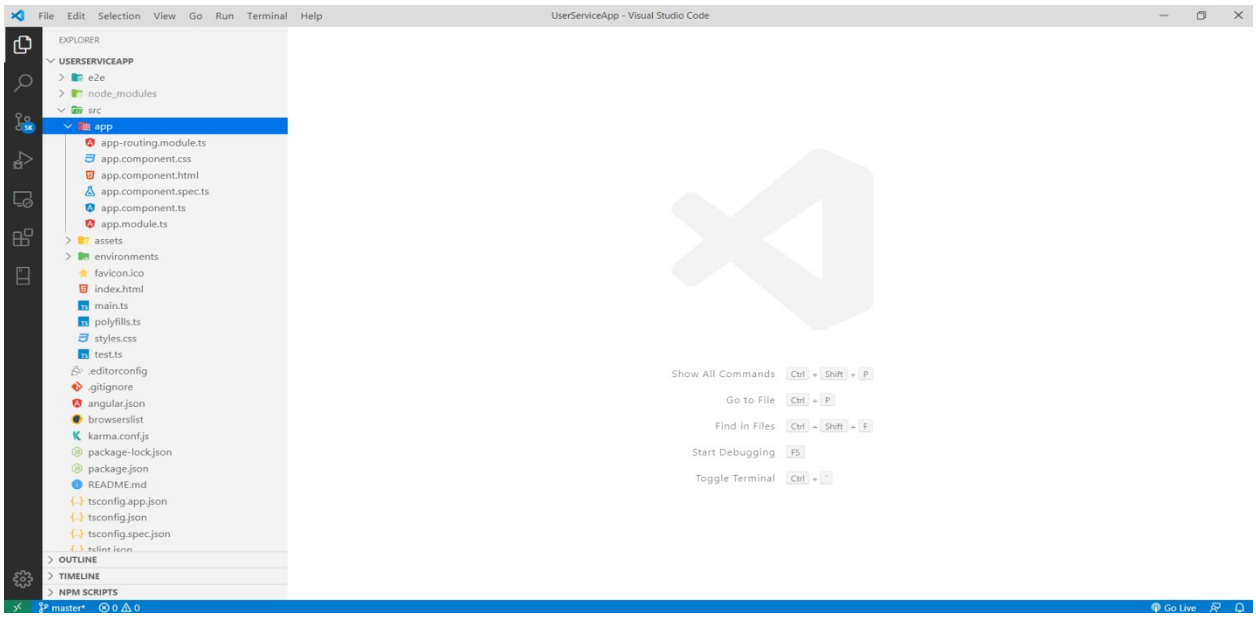
Steps to implement Service in Angular application

Step 1: First Create Angular Project by using following commands



Angular – Service Notes

Step 2: Open project into visual studio code (File -> open folder ->select project) after that look like following screen



Step 3: Add bootstrap code into index.html

->copy css from getbootstrap.com site or else use following one
https://getbootstrap.com/docs/4.5/getting-started/introduction/

Just copy css link and add into index.html head section

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.cs
s" integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">
```

->same kind of way copy js file and into index.html before closing body tag

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384
-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha3
84-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI" crossorigin="anonymous"></script>
```

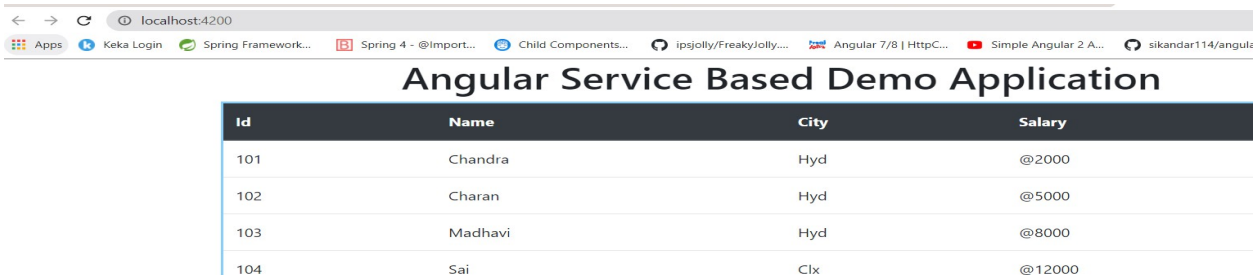
Step 4: To test user array elements accessible in to html or not checking purpose declare one users array into **app.component.ts**

Angular – Service Notes

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Angular Service Based Demo Application';

  users:any[]=[
    {id:101,name:'Chandra',city:'Hyd',salary:2000},
    {id:102,name:'Charan',city:'Hyd',salary:5000},
    {id:103,name:'Madhavi',city:'Hyd',salary:8000},
    {id:104,name:'Sai',city:'Clx',salary:12000},
    ];
}
```

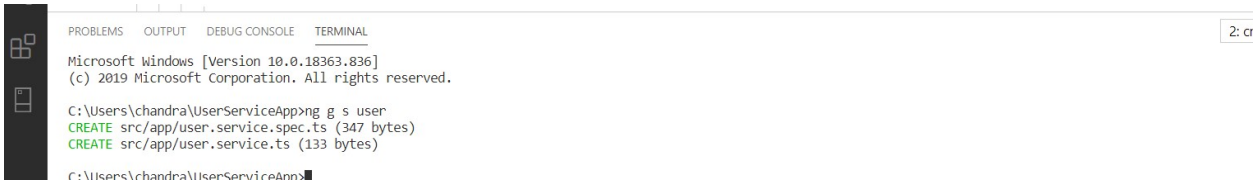
Step 4: To display array Elements into app.component.html use the following html table

```
<div class="container">
  <h1 class="text-center">
    {{title}}
  </h1>
  <table class="table" >
    <thead class="thead-dark">
      <tr>
        <th scope="col">Id</th>
        <th scope="col">Name</th>
        <th scope="col">City</th>
        <th scope="col">Salary</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let user of users">
        <td>{{user.id}}</td>
        <td>{{user.name}}</td>
        <td>{{user.city}}</td>
        <td>@{{user.salary}}</td>
      </tr>
    </tbody>
  </table>
</div>
```



Angular – Service Notes

Step 5: Create a service type script class to maintain shared resource (here user array)



Step 6: add shared resource array and method into service

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

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

  constructor() { }

  getAllUsers(){
    return [
     {id:101,name:'Chandra',city:'Hyd',salary:2000},
    {id:102,name:'Charan',city:'Hyd',salary:5000},
    {id:103,name:'Madhavi',city:'Hyd',salary:8000},
    {id:104,name:'Sai',city:'Clx',salary:12000},
    ];
    }
}
```

Step 7: To register service update in to app.module.ts file

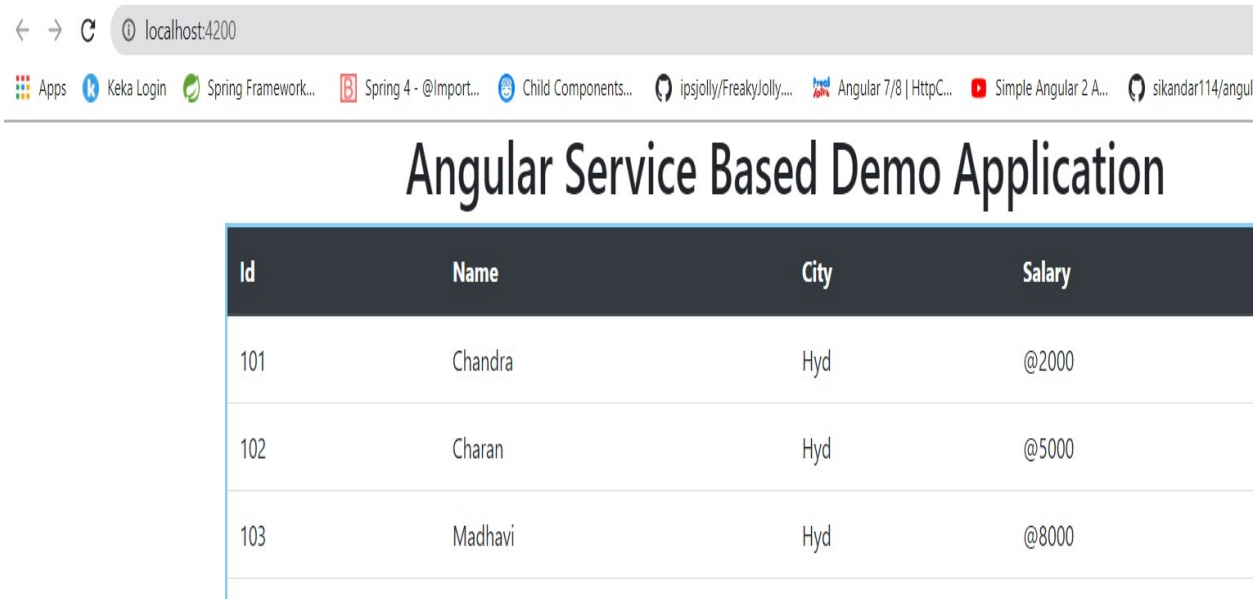
```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

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

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

Angular – Service Notes

Step 8: Now update app.component.ts to get the users array from service

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

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

  title = 'Angular Service Based Demo Application';

  constructor(private userService:UserService){}

  users:any[]=[
    {id:101,name:'Chandra',city:'Hyd',salary:2000},
    {id:102,name:'Charan',city:'Hyd',salary:5000},
    {id:103,name:'Madhavi',city:'Hyd',salary:8000},
    {id:104,name:'Sai',city:'Clx',salary:12000},
    ];

    ngOnInit(){
      this.users=this.userService.getAllUsers();
    }
}
```

Finally you can get same kind of output but service shared component you can use not only app.component.ts any other component also.



Angular – Service Notes