HTTP SERVICE

Prepare one component

### ng g c user

generate service

# ng g s dataservice

now two files will be prepared

1. data-service.service.ts
2. data-service.service.spec.ts

goto **data-service.service.ts**

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

import {HttpClient} from '@angular/common/http'

import { Observable } from 'rxjs';

import { UserClass } from './UserClass';

@Injectable({

  providedIn: 'root'

})

export class DataServiceService {

  //url:https://jsonplaceholder.typicode.com/users

  url:string ="https://jsonplaceholder.typicode.com/users";

  //inject DI

  constructor(private http:HttpClient) { }

  getAlluser():Observable<UserClass[]>{

    return this.http.get<UserClass[]>(this.url);

  }

}

**Right click on app>new file**

**UserClass.ts**

export class UserClass{

    id:number;

    name:string;

    username:string;

    email:string;

}

**For removing compile time error**

**Goto>** tsconfig.json **file add below line**

/\* To learn more about this file see: https://angular.io/config/tsconfig. \*/

{

  "compileOnSave": false,

  "compilerOptions": {

    "baseUrl": "./",

    "outDir": "./dist/out-tsc",

    "forceConsistentCasingInFileNames": true,

    "strictPropertyInitialization": false,

    "strict": true,

    "noImplicitOverride": true,

    "noPropertyAccessFromIndexSignature": true,

    "noImplicitReturns": true,

    "noFallthroughCasesInSwitch": true,

    "sourceMap": true,

    "declaration": false,

    "downlevelIteration": true,

    "experimentalDecorators": true,

    "moduleResolution": "node",

    "importHelpers": true,

    "target": "es2020",

    "module": "es2020",

    "lib": [

      "es2020",

      "dom"

    ]

  },

  "angularCompilerOptions": {

    "enableI18nLegacyMessageIdFormat": false,

    "strictInjectionParameters": true,

    "strictInputAccessModifiers": true,

    "strictTemplates": true

  }

}

**Now inject the service in** user.component.ts **file**

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

import { DataServiceService } from '../data-service.service';

import { UserClass } from '../UserClass';

@Component({

  selector: 'app-user',

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

  styleUrls: ['./user.component.css']

})

export class UserComponent implements OnInit {

  //inject the service

  constructor(private service:DataServiceService) { }

  users:UserClass[];

  ngOnInit(): void {

    this.service.getAlluser().subscribe(result=>this.users=result);

  }

}

User.component.html **file**

<h1>HTTP Service</h1>

<table class="table table-bordered table-striped">

    <thead>

        <tr>

            <th>ID</th>

            <th>NAME</th>

            <th>USERNAME</th>

            <th>EMAIL</th>

        </tr>

    </thead>

    <tbody>

        <tr \*ngFor="let u of users">

            <td>{{u.id}}</td>

            <td>{{u.name}}</td>

            <td>{{u.username}}</td>

            <td>{{u.email}}</td>

        </tr>

    </tbody>

</table>

**Now Add routes**

**App.routing.module.ts file**

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { AboutusComponent } from './aboutus/aboutus.component';

import { ContactusComponent } from './contactus/contactus.component';

import { HomeComponent } from './home/home.component';

import { UserComponent } from './user/user.component';

const routes: Routes = [

  //path to redirect

  {"path":"about",component:AboutusComponent},

  {"path":"home",component:HomeComponent},

  {"path":"contact",component:ContactusComponent},

  {"path":"user",component:UserComponent},

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

**app.component.html**

<h1>Routing Demo</h1>

<br>

<ul class="nav">

  <li class="nav-item">

    <a class="nav-link" routerLink="home">Home</a>

  </li>

  <li class="nav-item">

    <a class="nav-link" routerLink="about">About US</a>

  </li>

  <li class="nav-item">

    <a class="nav-link" routerLink="contact">Contact Us</a>

  </li>

  <li class="nav-item">

    <a class="nav-link" routerLink="user">User Http Service</a>

  </li>

</ul>

<router-outlet></router-outlet>

**Save and check the out put**



