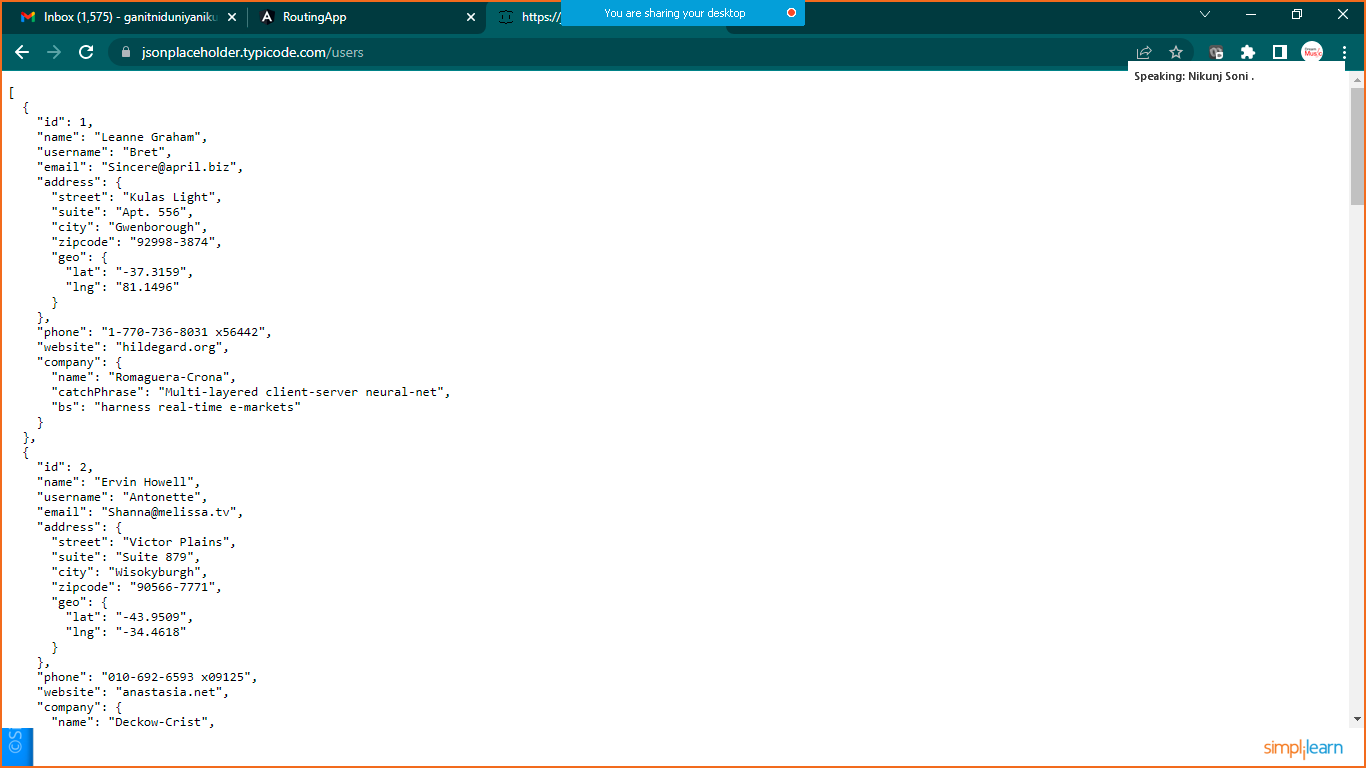
HTTP SERVICE

Http service is used to fetch API from other source to angular app

Let’s Get started

Basically we are going to fetch data from the below link

Link: <https://jsonplaceholder.typicode.com/users>



**Step:1 Generate New Component Called User**

* ng g c user

**Step:2 Generate Service**

* ng g s dataService

**Step:3 Prepare UserClass.ts file**

Goto> app>rightclick >new file> givename

**Step:4 Write down below code in UserClass.ts File**

export class UserClass{

    id:number;

    name:string;

    username:string;

    email:string;

}

**Step: 5 data-service.service.ts file import HttpClient Mannually from ‘@angular/common/http’**

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 the DI

  constructor(private http:HttpClient) { }

  getAllUsers(): Observable<UserClass[]>{

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

  }

}

**Step: 6 inject the service in User component**

Goto> **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.getAllUsers().subscribe(result=>this.users=result);

  }

}

Step: 7 Register The **HttpClientModule** in **app.module.ts** file

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

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

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

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

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

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

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

import { HttpClient, HttpClientModule } from '@angular/common/http';

@NgModule({

  declarations: [

    AppComponent,

    HomeComponent,

    AboutusComponent,

    ContactusComponent,

    UserComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Step: 8 implement the routing for user component

Goto> **approuting.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":"home",component:HomeComponent},

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

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

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

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

Step: 9 Implement the routerLink in Navigation on Home Page

Goto> **app.component.html** file

<h1>Routing Mechnism Demo</h1>

<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="aboutus">AboutUs</a>

  </li>

  <li class="nav-item">

    <a class="nav-link" routerLink="contactus">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 it and check the output

