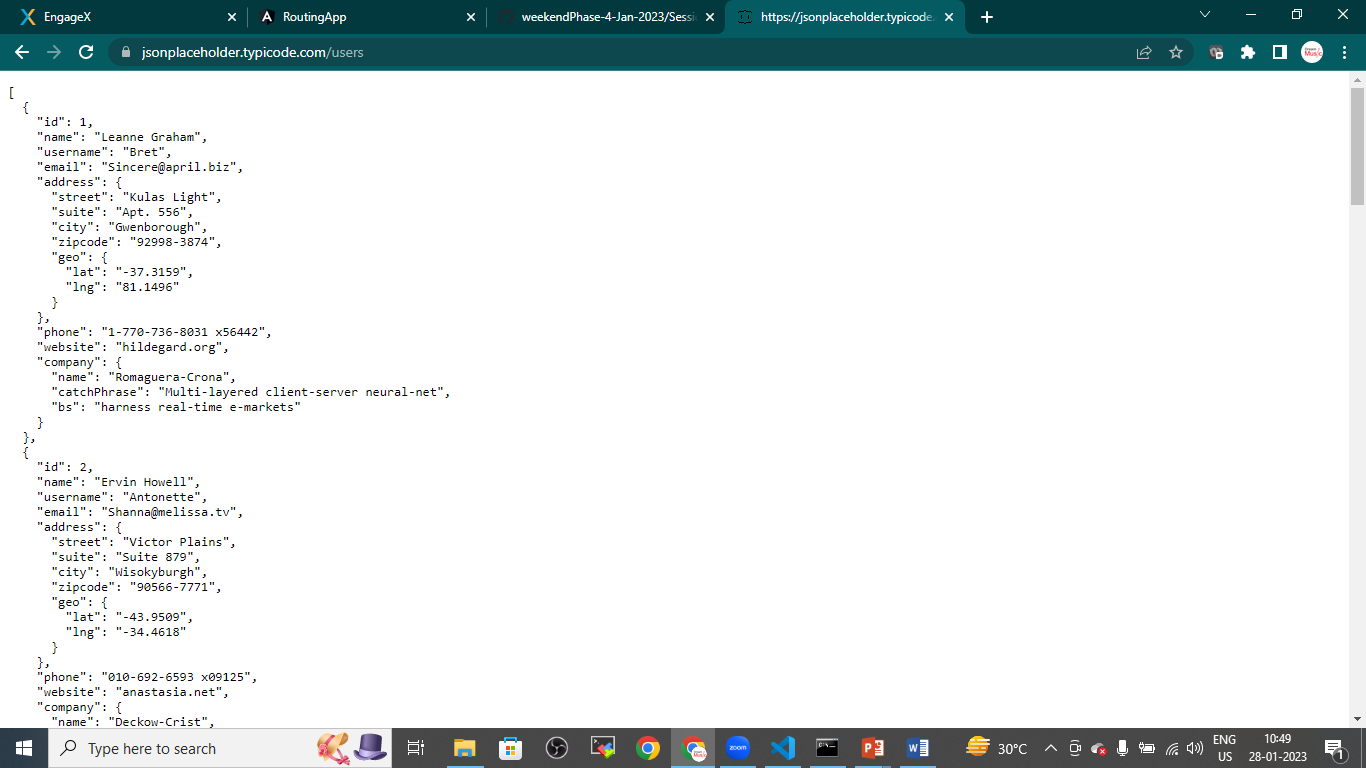
**Http Service**

**http service is used to fetch API from other source to angular app**

**let’s get started:**

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



**STEP 1: GENERATE NEW COMPONENT IN ANY EXISTING APP**

* ng g c Http

**STEP 2: Generate Service**

* ng g s data

**STEP 3: Generate UserClass.ts file and declare the DataTypes**

* Goto>src>app>rightclick >give name

**UserClass.ts** file

export class UserClass{

    id:number;

    name:string;

    username:string;

    email:string;

}

**Note:** if you are getting any compile time error in latest angular version **goto>tsConfig.json** File and below highlighted code

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

{

  "compileOnSave": false,

  "compilerOptions": {

    "baseUrl": "./",

    "outDir": "./dist/out-tsc",

    "strictPropertyInitialization":false,

    "forceConsistentCasingInFileNames": true,

    "strict": true,

    "noImplicitOverride": true,

    "noPropertyAccessFromIndexSignature": true,

    "noImplicitReturns": true,

    "noFallthroughCasesInSwitch": true,

    "sourceMap": true,

    "declaration": false,

    "downlevelIteration": true,

    "experimentalDecorators": true,

    "moduleResolution": "node",

    "importHelpers": true,

    "target": "ES2022",

    "module": "ES2022",

    "useDefineForClassFields": false,

    "lib": [

      "ES2022",

      "dom"

    ]

  },

  "angularCompilerOptions": {

    "enableI18nLegacyMessageIdFormat": false,

    "strictInjectionParameters": true,

    "strictInputAccessModifiers": true,

    "strictTemplates": true

  }

}

STEP 5: Import HttpClient and Observable Manually from ‘@agulat/common/http’ in DataService.ts file

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

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

import { Observable } from 'rxjs';

import { UserClass } from './UserClass';

@Injectable({

  providedIn: 'root'

})

export class DataService {

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

  //inject DI

  constructor(private http:HttpClient) {  }

  getAllUsers():Observable <UserClass[]>{

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

  }

}

STEP 6: Inject the service in HttpComponent.ts file

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

import { DataService } from '../data.service';

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

@Component({

  selector: 'app-http',

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

  styleUrls: ['./http.component.css']

})

export class HttpComponent implements OnInit{

  //inject the service

  constructor(private service:DataService){}

  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 { ContactComponent } from './contact/contact.component';

import { HttpComponent } from './http/http.component';

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

@NgModule({

  declarations: [

    AppComponent,

    HomeComponent,

    AboutusComponent,

    ContactComponent,

    HttpComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

STEP 8: Prepare the navigation in app.component.html file

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

        <a href="#" class="navbar-brand">Simplilearn</a>

        <ul class="navbar-nav">

            <li class="nav-item">

                <a href="#" class="nav-link" routerLink="home">HOME</a>

            </li>

            <li class="nav-item">

                <a href="#" class="nav-link" routerLink="about">ABOUT US</a>

            </li>

            <li class="nav-item">

                <a href="#" class="nav-link" routerLink="contact">CONTACT US</a>

            </li>

            <li class="nav-item">

              <a href="#" class="nav-link" routerLink="http">HTTP SERVICE</a>

          </li>

        </ul>

    </nav>

    <router-outlet></router-outlet>

Step 9: Give routes to the component in app-routing.component.ts file

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

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

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

import { ContactComponent } from './contact/contact.component';

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

import { HttpComponent } from './http/http.component';

const routes: Routes = [

  //path to redirect

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

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

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

  {"path":"http",component:HttpComponent},

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

STEP 10: Prepare Html file (Table) to Load User Data in http.component.html

<hr>

<h2>User Details</h2>

<hr>

<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>

Save it and run the App and check the output

