Services and Injectables Demo

Create an application: ng new depinj

Move: cd depinj

Open folder in VS Code: code .

Now run: ng serve, check output localhost:4200

We want to call an api not directly from component but from service.

What the api: <https://jsonplaceholder.typicode.com/users>

I want to do the call of this api in my application

Let’s create a Model in angular Application

Right click on app and create new file> user.ts

export class User{

    id: number;

    name:string;

    email:string;

    username:string;

    address:Address;

    phone:string;

    website:string;

}

export class Address{

    street:string;

    suite:string;

    city:string;

    zipcode:string;

    geo:Geo

}

export class Geo{

    lat:number;

    lng:number;

}

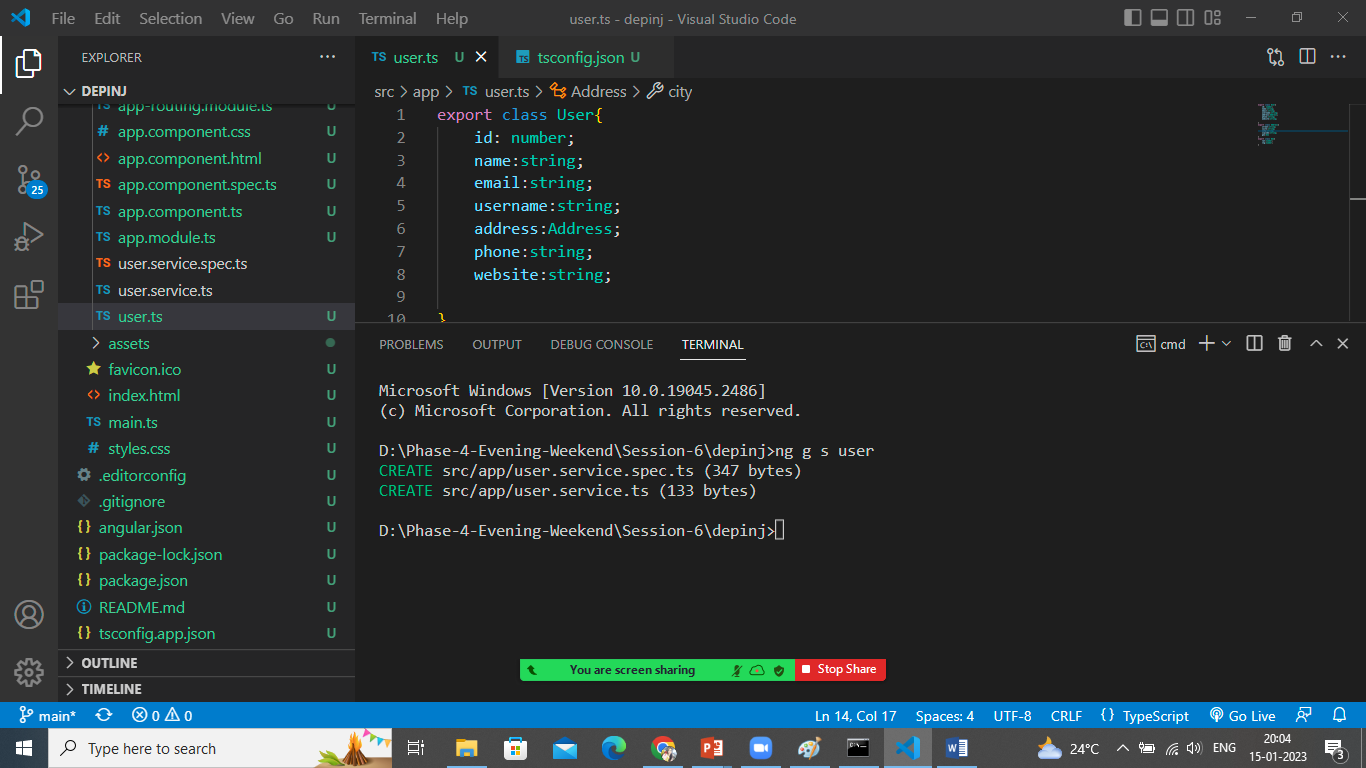
To solve the error you need to edit tsconfig.json file

"compilerOptions": {

    "strictPropertyInitialization": false,

Save this and the error will get resolved.

To do the api calls create service: ng g s user



To do the api calls and to call the HTTP methods we have to import one module in app.module.ts file

HTTPClientModule

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

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

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

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

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

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Open user.service.ts file

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

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

import {Observable} from 'rxjs';

import { User } from './user';

@Injectable({

  providedIn: 'root'

})

export class UserService {

  //Let's inject Http Dependency to make an API Calls

  constructor(private http:HttpClient) { }

  apilink:string="https://jsonplaceholder.typicode.com/users"

  //users array but its real time

  getAllUsers():Observable<User[]>{

    return this.http.get<User[]>(this.apilink);

  }

  getSingleUser(id:number):Observable<User>{

    return this.http.get<User>(this.apilink+"/"+id);

  }

  //normal calling api

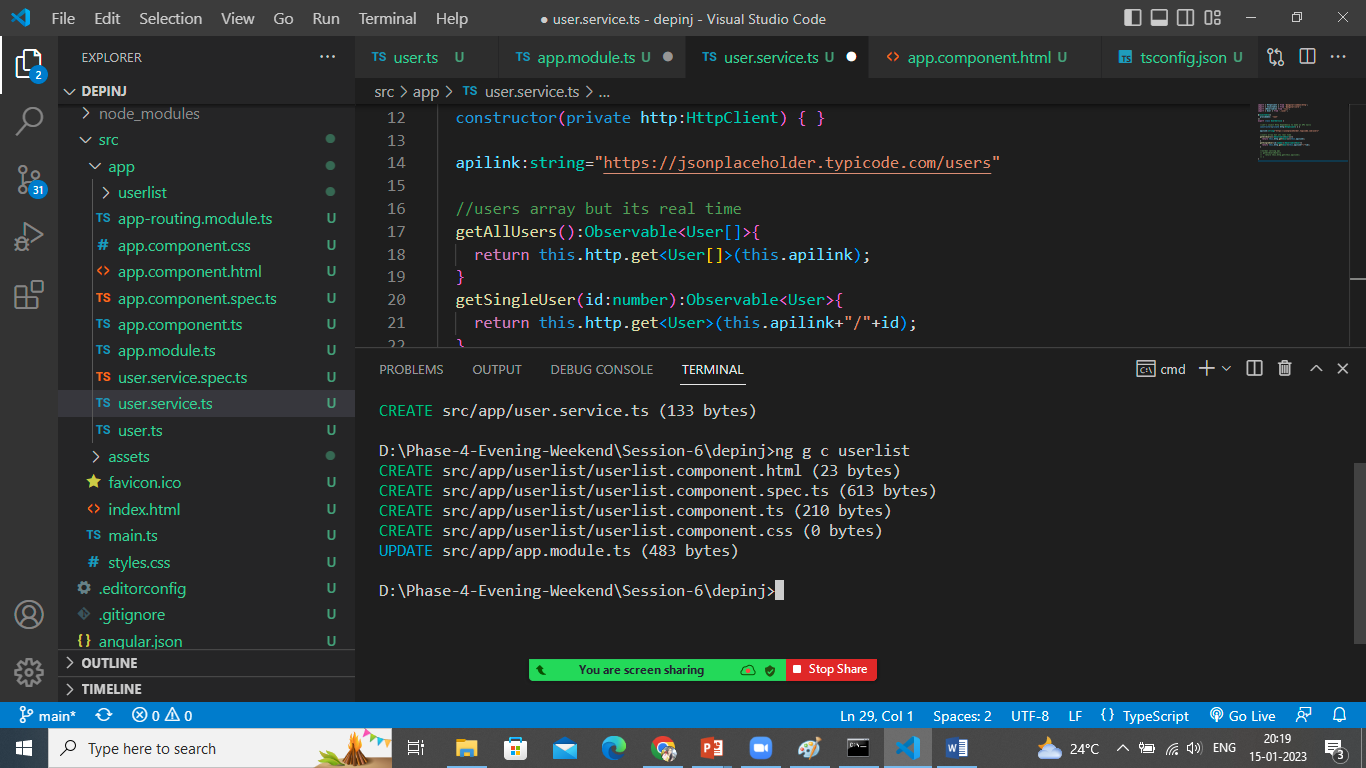
  // getAllUsersData(){

  //   return this.http.get(this.apilink);

  // }

}

To call the service in component let’s create a component



Open userlist.component.ts

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

import { User } from '../user';

import { UserService } from '../user.service';

@Component({

  selector: 'app-userlist',

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

  styleUrls: ['./userlist.component.css']

})

export class UserlistComponent {

  list:User[]

  //injected the service dependency in Component

  constructor(private service: UserService){ }

  ngOnInit(){

    this.service.getAllUsers().subscribe(x=>this.list=x)

    //when you call this method it is returning observable so you need to

    //subscribe the data and the received data store it in a variable list

  }

}

Print data on HTML page

Userlist.component.html

<h1>Users Data</h1>

<table>

    <thead>

        <tr>

            <th>Id</th>

            <th>Name</th>

            <th>UserName</th>

            <th>Email</th>

            <th>Address</th>

        </tr>

    </thead>

    <tbody>

        <tr \*ngFor="let u of list">

            <td>{{u.id}}</td>

            <td>{{u.name}}</td>

            <td>{{u.username}}</td>

            <td>{{u.email}}</td>

            <td>{{u.address.state}}, {{u.address.city}},{{u.address.geo.lat}}</td>

        </tr>

    </tbody>

</table>

Insert this component on app.component.html

<div class="container">

  <app-userlist></app-userlist>

</div>

Check the o/p



To use same instance of your service everytime you can register your service in providers array in app.module.ts file

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

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

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

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

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

import { UserlistComponent } from './userlist/userlist.component';

import { UserService } from './user.service';

@NgModule({

  declarations: [

    AppComponent,

    UserlistComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule

  ],

  providers: [UserService],

  bootstrap: [AppComponent]

})

export class AppModule { }

save all the files.