Service is class that can send functionality and data across multiple components

Ng g service services/email //moje vmesto service samo s da napisha

Creating services

//1 pravim si folder I typescript fajl /services/my-service.service.ts

//2 import @Injectable I pravim class, koeto ni pozvoliava da vzemem tozi service I da go injektnem kato component

//3 advame go kat provider v @ngModule

//4 call the component

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

import {User} from './user';

import {USERS} from './mock-users';

@Injectable()

export class UserService {

getUsers(): User[]{

return USERS;

}

}

//componenite gi dobaviame v declaration; services gi dobaviame v providers v app.module.ts

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

@NgModule({

declarations: [

AppComponent,

SandboxComponent

],

imports: [

BrowserModule,

FormsModule

],

providers: [DataService],

bootstrap: [AppComponent]

})

export class AppModule { }

//kogato polzvame services triabva da ia injectnem kato dependency, a tuy stave v ctora

//v component ts

/1vo importvame Component clasa

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

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

@Component({

selector: 'sandbox',

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

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

})

export class SandboxComponent {

users:string[];

//injektvame kat dependency;pravim ia da e sus sushtoto ime :DataService kat importnata

constructor(public dataService: DataService){

//zimame vsichko koeto imame v samia service

this.users = this.dataService.getUsers();

}

}

I posle se displayva v html s foreach

**Observables** – open lines of communications pr open data stream, koeto gi pravi mnogo burzi

V service-a si importvame

import { Observable } from 'rxjs';

@Injectable()

export class DataService {

//definirame si prop data ot tip observable koiato e array ot numbers

data: Observable<Array<number>>;

constructor(){

}

getData(){

this.data = new Observable(ob => {

setTimeout(() => {

ob.next(‘hi’);

}, 1000);

setTimeout(() => {

ob.next(2);

}, 2000);

setTimeout(() => {

ob.complete();

}, 5000);

});

//posle v componenta, v ctora triabva da se subscribnem

export class SandboxComponent {

//definirame si prop ot tip array any

data: any[] = [];

constructor(public dataService: DataService){

//tuy kato e observable triabva da se subscribnem kum neia

//to vrushta kakvoto idva ot data

this.dataService.getData().subscribe(data => {

console.log(data);

});

}

}

HTTP module and observable

//dobaviame si module-a kum app.module.ts

import { HttpModule } from '@angular/http';

//dobaviame go I v import sekciata

imports: [

BrowserModule,

FormsModule,

HttpModule

],

//v service si importvame http module-a

import { Http } from '@angular/http';

//dobaviame si map

import 'rxj/add/operator/map'; //pozvoliava da polzvame map

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

import { Observable } from 'rxjs';

import { Http } from '@angular/http';

import 'rxjs/add/operator/map'; //pozvoliava da polzvame map

@Injectable()

export class DataService {

constructor(public http: Http){

}

getUsers(){

//shte vurne json datata ot apito

//polzvame map shtoto iskame da returnem observable

//triabva da mapnem response-a kum json predi da go izpratim

return this.http.get('http://jsonplaceholder.typicode.com/users')

.map(res => res.json()); //tuka sveti shtoto bibliotekata rxjs ne mi e pravilna za versiata na angular

}

}

//componenta ni e taka:

//1vo importvame Component clasa

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

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

@Component({

selector: 'sandbox',

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

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

})

export class SandboxComponent {

users: any[] = [];

constructor(public dataService: DataService){

this.dataService.getUsers().subscribe(users => {

// console.log(users);

//pravim si userite da sa = na userite det se vrushtat ot service-a

this.users = users;

})

}

}

//I posle v html/a niakuv for si pravim

**HTTP POST**