**Angular - Http Service**

Http Service will help us fetch external data, post to it, etc. We need to import the http module to make use of the http service. Let us consider an example to understand how to make use of the http service.

To start using the http service, we need to import the module in **app.module.ts** as shown below −

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

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

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

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

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

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

BrowserAnimationsModule,

HttpModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

If you see the highlighted code, we have imported the HttpModule from @angular/http and the same is also added in the imports array.

Let us now use the http service in the **app.component.ts**.

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

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

import 'rxjs/add/operator/map';

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

constructor(private http: Http) { }

ngOnInit() {

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

map((response) ⇒ response.json()).

subscribe((data) ⇒ console.log(data))

}

}

Let us understand the code highlighted above. We need to import http to make use of the service, which is done as follows −

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

In the class **AppComponent**, a constructor is created and the private variable http of type Http. To fetch the data, we need to use the **get API** available with http as follows

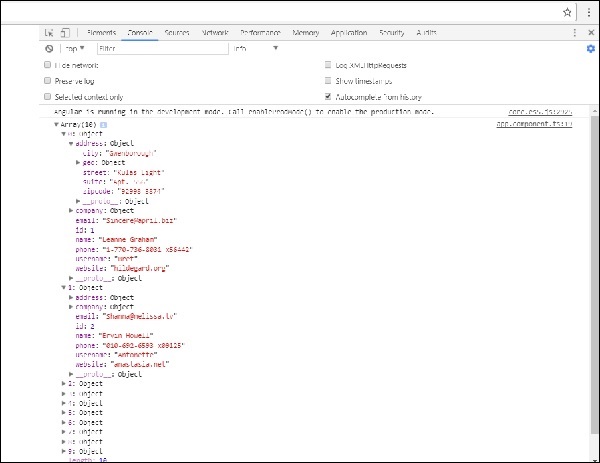
this.http.get();

It takes the url to be fetched as the parameter as shown in the code.

We will use the test url - <https://jsonplaceholder.typicode.com/users> to fetch the json data. Two operations are performed on the fetched url data map and subscribe. The Map method helps to convert the data to json format. To use the map, we need to import the same as shown below −

import 'rxjs/add/operator/map';

Once the map is done, the subscribe will log the output in the console as shown in the browser −



If you see, the json objects are displayed in the console. The objects can be displayed in the browser too.

For the objects to be displayed in the browser, update the codes in **app.component.html** and **app.component.ts** as follows −

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

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

import 'rxjs/add/operator/map';

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

constructor(private http: Http) { }

httpdata;

ngOnInit() {

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

map(

(response) ⇒ response.json()

).

subscribe(

(data) ⇒ {this.displaydata(data);}

)

}

displaydata(data) {this.httpdata = data;}

}

In **app.component.ts**, using the subscribe method we will call the display data method and pass the data fetched as the parameter to it.

In the display data method, we will store the data in a variable httpdata. The data is displayed in the browser using **for** over this httpdata variable, which is done in the **app.component.html** file.

<ul \*ngFor = "let data of httpdata">

<li>Name : {{data.name}} Address: {{data.address.city}}</li>

</ul>

The json object is as follows −

{

"id": 1,

"name": "Leanne Graham",

"username": "Bret",

"email": "Sincere@april.biz",

"address": {

"street": "Kulas Light",

"suite": "Apt. 556",

"city": "Gwenborough",

"zipcode": "92998-3874",

"geo": {

"lat": "-37.3159",

"lng": "81.1496"

}

},

"phone": "1-770-736-8031 x56442",

"website": "hildegard.org",

"company": {

"name": "Romaguera-Crona",

"catchPhrase": "Multi-layered client-server neural-net",

"bs": "harness real-time e-markets"

}

}

The object has properties such as id, name, username, email, and address that internally has street, city, etc. and other details related to phone, website, and company. Using the **for** loop, we will display the name and the city details in the browser as shown in the **app.component.html** file.

This is how the display is shown in the browser −



Let us now add the search parameter, which will filter based on specific data. We need to fetch the data based on the search param passed.

Following are the changes done in **app.component.html** and **app.component.ts** files −

### app.component.ts

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

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

import 'rxjs/add/operator/map';

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'app';

searchparam = 2;

jsondata;

name;

constructor(private http: Http) { }

ngOnInit() {

this.http.get("http://jsonplaceholder.typicode.com/users?id="+this.searchparam).

map(

(response) ⇒ response.json()

).

subscribe((data) ⇒ this.converttoarray(data))

}

converttoarray(data) {

console.log(data);

this.name = data[0].name;

}

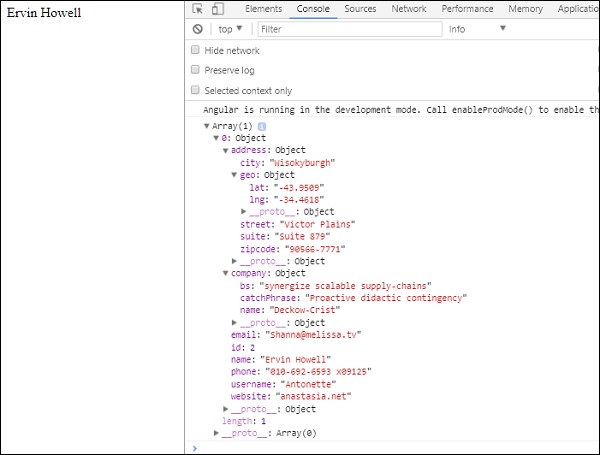
}

For the **get api**, we will add the search param id = this.searchparam. The searchparam is equal to 2. We need the details of **id=2** from the json file.

### app.component.html

{{name}}

This is how the browser is displayed −



We have consoled the data in the browser, which is received from the http. The same is displayed in the browser console. The name from the json with **id=2** is displayed in the browser.