

HttpClient in Angular

Angular application can make HTTP requests to external data resources like Web/HTTP servers, File systems, APIs etc. Then it will receive the data as response and it can consume it.

To make HTTP Request from Angular, We need object of HttpClient class which was available in @angular/common/http package. This class is available as service.

To get object of HttpClient class in to our angular app, do the following two steps

Import “HttpClientModule” from @angular/common/http to AppModule

Add module name to imports array

Because of the above two steps, the HttpClient object is provided into root injector of application. With the help of it, HTTP calls can be made. The following methods are available in HttpClient class.

HTTP Req Type	Description	HttpClient method
GET	To read data	get(url:string, options:object):Observable<ArrayBuffer>
POST	To create data	post(url:string, data:any, options:object):Observable<ArrayBuffer>
PUT	To modify data	put(url:string, data:any, options:object):Observable<ArrayBuffer>
PATCH	To partial updates	patch(url:string, data:any, options:object):Observable<ArrayBuffer>
DELETE	To delete data	delete(url:string, options:object):Observable<ArrayBuffer>

Use services to make HTTP requests

An HTTP call can be made either from component or service. But it is more useful when we make call from service than component based on the following benefits.

Separation of HTTP calls logic from components

To use components exclusively for view-related logic.

Observable

All methods of HttpClient return Observable as result. It is a special mechanism to deal with Asynchronous data. When the client makes a HTTP request, its result may arrive in the future once the request processing is done by the http server. So it belongs to blocking kind of request. To deal with this blocking requests efficiently, asynchronous programming is developed.

Let us make HTTP GET request from a service to url "<https://reqres.in/api/users?page=2>" which provides sample data to us. (You can take other URLs to test)

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class DataService {

  //make a http get req
  //inject HttpClient obj
  constructor(private hc:HttpClient) { }

  //make req
  makeAHttpReq():Observable<object>
  {
    //here the url "https://reqres.in/api/users?page=2" will send an object,
    //request type of object(get<object>()) and returns Observable of type object

    return this.hc.get<object>("https://reqres.in/api/users?page=2");
  }
}
```

Getting data from Observable

Once a service is provided with HTTP request making mechanism, it will return Observable as the result of that call. If a component needs data from that observable, that component has to subscribe to that Observable by calling following method.

Observable-object. subscribe(function)

Observables are lazy by nature. That means, until they subscribed, the HTTP call will not be completed.

The "subscribe()" can read data from Observable object and pass it as argument to its parameter function.

Consider the following TestComponent can subscribe the observable of above DataService.

```

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

test.component.ts

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

@Component({
  selector: 'app-aboutus',
  templateUrl: './test.component.html',
  styleUrls: ['./test.component.css']
})
export class TestComponent implements OnInit {

  receivedData:object[]=[];

  //inject DataService obj
  constructor(private ds:DataService) { }

  ngOnInit(): void {
    //subscribe to the Observable returned by "makeAHttpReq()" " method
    this.ds.makeAHttpReq().subscribe((result)=>{
      //let us extract value of "data" key of "result" object
      this.receivedData=result["data"];
    });
  }
}

```

Now let us interpolate data to its view

```

Test.component.html
-----
<table class="table">
  <thead>
    <th>ID</th>
    <th>Email</th>
    <th>First Name</th>
    <th>Last name</th>
    <th>Profile photo</th>
  </thead>

  <tr *ngFor="let obj of receivedData">
    <td>{{obj.id}}</td>
    <td>{{obj.email}}</td>
    <td>{{obj.first_name}}</td>
    <td>{{obj.last_name}}</td>
    <td>
      <img [src]=obj.avatar alt="">
    </td>
  </tr>
</table>

```