Techiediaries

Ang 6|7: HttpClient —Bu g a Service for Sen API Calls and Fet Data

Re rials.

Your e-mail

Subscribe

Subscribe to receive new Angular 7 tutorials

Email Address

Subscribe

Throughout this tutorial, you'll
requests or make API calls to R

You can use the example code t
CORS enabled or to your back-
server tiers.

> *Typed ar l Headers in Angular 7 HttpClient: Link Header Paginati*
>
> *Angular rogress Bar Tutorial and Example*

Prerec

To be able to (first need to have a few requirements:

- Node.js and npm installed. If that's not the case simply go to the official website and download the latest version for your operating system,
- A back-end server with a REST API with CORS enabled.

You also need to have the Angu

```
$ npm install -g @angul
```

> *You may need to add sudo c*

If you don't have an Angular 7

```
$ ng start angular7-exa
```

Wait for the C : files and install the required dependencies then you are good to go!

The A pClient Module

Angular is a c ıting client side mobile and desktop apps. As such, you don't have to use other ex common operations like HTTP requests.

Angular provi odule which allows developers to send HTTP requests and make API calls to re

`HttpClient` `ngular/common/http` package and it replaces the old HTTP client that was `gular/http` package.

In web browse l APIs for sending HTTP requests which are the `XMLHttpRequ` `etch()` API (available only on modern browsers).

The `HttpCli` op of the `XMLHttpRequest` interface. It wraps all the complexities c des extra features like:

- RxJS O
- Intercep onses,
- Types fo ,
- Better e
- Support for testing,
- And so on.

In the next step, you are going to setup the `HttpClient` module in your Angular 7 project.

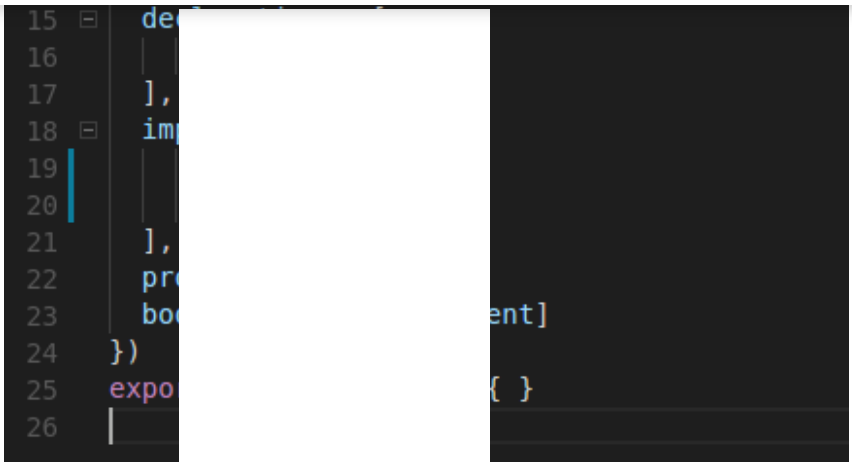
Configuring T

After introducing `HttpClient`

In fact, you don't have to do mu `@angular/common/http` pack

Open the `src/app/app.modul`

```
import { HttpClientModu
```



That's it! You ient library to send HTTP requests to a third-party API server or to your back-e

In the next ste : a service that will encapsulate the code that communicates with your server.

Creati ular 7 Service

After setting u project, you need to create a service which will import `HttpClient` TTP requests needed in your application.

This service w y component that needs to do any HTTP operations.

Go to your ter ting a service using the Angular CLI v7:


```
ng g service api
```

This will generate the `api.ser` that contains the service's code.

> *You will not add any tests in `api.service.ts` file.*

In the example, we suppose tha `www.server.com/api/` customer has the following attr

- id,
- firstName,
- lastName,

In the next ste or your data (customer).

Creati gular Model(s)

At this point, t ed to do is to create a `Customer` class which will be used as a type for each

> *The `Cust` odel.*

You need to st `stomer` model using the following command:

```
ng genera
```

> *You can al `generate` .*

This will gene the `src/app` folder

> *For simple on we used is OK but for big projects you may need to create a `models` your class model(s). And also use feature modules to encapsula project. For example, in this project you could use a `customer` the API service, the Customer model and any components then import it i odule.*

Next, open the `src/app/customer.ts` file and add:

```
export class Customer{
    id: number;
    firstName: string;
    lastName: string;
    email: string;
    phone: string;
    city: string;
}
```

> *The `Customer` class is a u*
> *`number` and `string` to d*
> *You also need to export the*
> *imported from other classes*

```
 import {                    /angular/core';
import { H                   ngular/common/http';

@Injectab
  provide
})
export cl
  apiURL:                    www.server.com/api/';

  construc                   ent: HttpClient) {}
}
```

These are the r opening the service file:

You first injec ate `httpClient` instance in the service's constructor.

You next add ble which stores the address of the remote API server.

In the next ste for doing create, read, update and delete operations against the the `www.serv  omers` endpoint.

Creati Methods

After creating he class model and injected `HttpClient` into the service, you can now create the CRUD methods

Since these methods will make use of the `Customer` model class as a type, either for the parameters or the return results, you first need to import the class inside the service's file:

```
 import { Customer } fro
```

Next, you can define the follow customers:

```
 public createCustomer(c

public updateCustomer(cu

public deleteCustomer(id
```

In the next sec l these methods one by one starting with the `.getCustomers` method which getting pages of data from the server.

Next, you'll in

- The `.c` which takes a parameter of the `Customer` type and send it to the server.
- The `.u` to update a customer,
- The `.d` to delete a customer by id,
- The `.g` d for getting a customer by id.

Fetchi ed Data from The API Servei

The `.getCus` used for fetching customers.

In a real-worl implement the logic for fetching paginated sets of data so in this example, you' ing information from the headers of the responses coming from the server.

For storing th 'll need to add the following variables in your service:

- `firstF` of the first page,
- `prevPage` for storing the URL of the previous page,
- `nextPage` for storing the URL of the next page,
- `lastPage` for storing the URL of the last page.

> *The `prevPage` and `nextP`*

In your `ApiService` add the f

```
public firstPage: strin
public prevPage: string
public nextPage: string
public lastPage: string
```

Now, you are ready to impleme simply be:

```
}
```

This will retur `omer[]>` that you need to subscribe to, in your components, in order to fetch the server.

> *We use the interpolation operator and back-ticks to format the endpoint URL.*

I you want to rovided that your server is implementing pagination using `Link` headers, you r de instead:

```
public ring){

  if(ur
    ret get<Customer[]>(url,{ observe: 'response' }).pipe(tap(r
      t on_links(res);
    }))
  }

  retur <Customer[]>(`${this.apiURL}/customers?page=1`,
  { obs ipe(tap(res => {
    thi _links(res);
  }));
}
```


Let's explain this code. You pass an optional `url` parameter to the method then inside the method body, you check if the `url` was passed:

- If an URL is passed to th
- If no parameter is passed
  `www.server.com/`

This way, you can use the `.ge` next pages by providing the UR received response.

Now, you need to pay attention

- First, you pass the `{obs` method. This tells `HttpC` us to retrieve the `Link` header that contains the paging information from the server.

Next, you nee ods in your service that will be used to retrieve pagination links:

- The `.p` hod to parse the `Link` header,
- The `.r` inks method to set pagination links once the `Link` header is parsed.

This is the imp se_link_header method:

```
  parse_l
    if (he
      retu
    }

    let pa ,');
    var l:
    parts
      let ');
      var olace(/<(.*)>/, '$1').trim();
      var eplace(/rel="(.*)"/, '$1').trim();
      linl

    });
    return
  }
```


This is the implementation of the `.retrieve_pagination_links` method:

```
public retrieve_paginat
    const linkHeader =
    this.firstPage =
    this.lastPage =
    this.prevPage =
    this.nextPage =
}
```

That's it. You have completed t

In the next section, you'll imple

Fetching a Single Object by its Identifier

single custom

```
public ge                    mber){
    retur                    :(`${this.apiURL}/customers/${id}`);
}
```

> *In this me* *he URL to a single customer by its id.*

In the next sec e method for creating a customer on the server.

Sendi Request to the Server

In this section `Customer` method which sends a POST request to the server. It takes a param

```
public cr                    r: Customer){
    retur                    st(`${this.apiURL}/customers/`,customer);
}
```

This method c `st` method that takes the URL and the customer object to send to the server wit

In the next sec date data on the server using `HttpClient` .

Sending a PUT Request to the Server

After creating the method to create customers on the server, you will create the `.updateCustomer` method that will be used to upd

```
public updateCustomer(c
    return this.httpCli
}
```

This method calls the `HttpCli`

In the next section, you'll add t the API server.

The last metho[illegible] is the `.deleteCustomer` method which deletes a single customer by its identifie

```
  public de                    ber){
    return                     lete(`${this.apiURL}/customers/${id}`);
  }
```

This method s[illegible] to the server using the `HttpClient.delete` method which takes an URL paran[illegible]ource you want to delete.

Testin[illegible] Methods

After creating [illegible] encapsulates all the methods to send HTTP requests to the API server. Now, l[illegible]th a simple example. Keep in mind that you need to have an HTTP server with an[illegible]oint.

If you have th[illegible]p.component.ts` file and start by importing `ApiService` then inject it in `Ap[illegible]`

```
 import {                    from '@angular/core';
import { A                  'api.service';

@Component(
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent{

  constructor(private ap
}
```


Next, add the following code in[illegible]

```
   ngOnInit(){
    this.apiService.getC
      this.apiService.ge
        console.log(res
      });
    });
```

Once the first can get the URL of the next page of data from `apiService.` use it as a parameter to the second call of the `.getCustomers` method to retr so on.

This can be id ding first, previous, next and last buttons that call the `.getCustome` esponding URLs that can be accessed from the `this.apiSer` `s.apiService.previousPage` , `this.apiService.nextPage` and `this.ap` riables.

Next, create a

```
var c
  "id'
  "fi                    e",
  "la
  "em                    n",
  "ph                    3",
  "ci
  "co
  "ti
}
```

And use it to c erver:


```
this.apiService.createCustomer(customer).subscribe((res)=>{
     console.log("Created a customer");
});
```

You can also update the custom

```
this.apiService.update(
  console.log("Updated
});
```

Finally, you can delete the cust

```
this.apiService.delete(
     console.log("Deleted a customer");
});
```

> *returned fi*

Conclu

In this tutorial ngular 7 and `HttpClient` to send HTTP requests to the server.

11 Nov 2018

- angula

« useState Re: :Angular 6|7: JWT Authentication Tutorial »

Author

Team

Techiediaries is a website dedicated to bring you tutorials for the latest web technologies

RELAT IALS

Content Pro in Angular 8 with ng-content

Angular 8 n plateOutlet Example

Lazy Loade n Angular 8 with loadChildren & Dynamic Imports

Building an rm in Angular 8

Understandi ntRef by Example

Angular 8 D hild and ViewChildren Example

Angular 8 U pdate & Angular CLI v8

Subscribe ir

What do you think?

35 Responses

John Hammond • 5 months ago • edited

Hi great article,

Using Observable<http
.subscibe(x =>...) to be
This creates a tight cou

There is an Angular libr
with strongly-typed call
With this, you only deal

https://github.com/Ver

Reply • Share

FibreTiger.Co.Za • 6 month

FOR THE LOVE OF GC
are like 100 files in an A
referencing !

Sir, I tested with json file and it succesfully worked. But I need to create a rest api call querying with oracle database, how to store queried data in json object? Please help

• Reply • Share ›

techiediaries.com Mod ➚ Katheeja Beevi • 7 months ago

This depends on the server side framework you are using. Angular is just for the front-end. Are you using PHP, Python Nods.js for the server?

• Reply • Share ›

Katheeja Beevi ➚ techiediaries.com • 7 months ago

No sir, I am using node js and javascript.

• Reply • Share ›

Show more replies

ALSO ON TECHIEDIARIES

Angular 7 File Upload with Progress Bar Tutorial and Example

1 comment • 4 months ago

Alex Schirmer — Good Job! simple explain

How to Post FormData (multipart/form-data) with Angular 7 and HttpClient

1 comment • 4 months ago

SK — When I m choosing the file i 'm getting this error:-ERROR DOMException: Failed to set the 'value' property on 'HTMLInputElement': This ...

Ionic 4 React Tutorial: Build a Mobile App with Ionic 4, Axios and React

1 comment • 3 months ago

Matt — Thanks! I disabled the axios data in this screenshot, but I followed your layout. The header isn't quite right yet on the iOS devices ...

PHP Image/File Upload Tutorial and Example [FormData and Angular 7 Front-End]

5 comments • 4 months ago

Zeno — Hey there, thanks for the great tutorial.Is it possible to do the same with template driven forms?Or is it possible to mix ...

Custom Search