**Communicating with backend services using http**

Step 1:- Add HttpClientModule

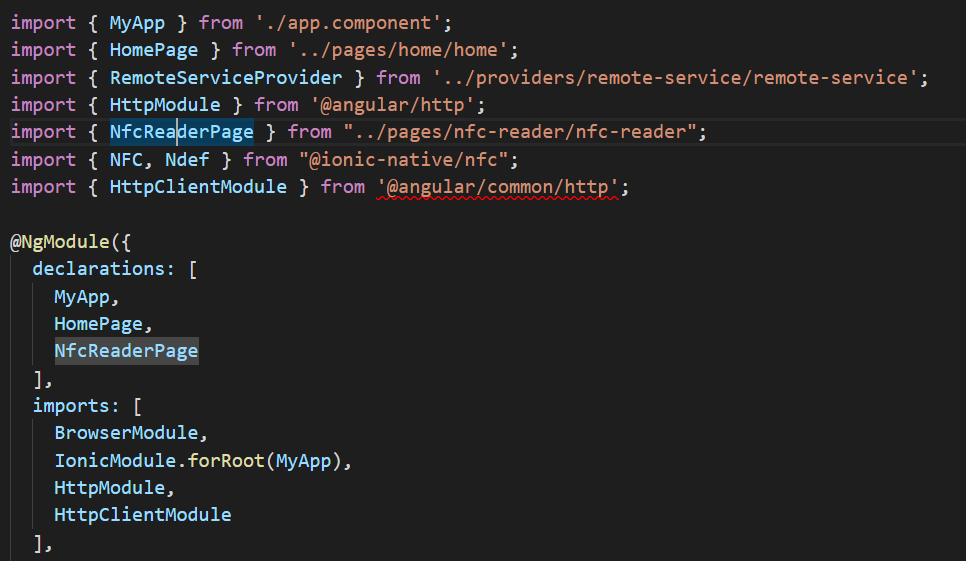
File name: - app.module.ts –

Import { HTTPClientModule } from ‘@angular/common/http’;

@NgModule({

Imports: [HttpClientModule],

})



Step 2:- Add HttpClient

File name:- app.component.ts

Import { HTTPClient } from ‘@angular/common/http’;

Export class AppComponent {

//Need to mention root URL variable http end point that we going to use

Readonly ROOT\_URL = ‘https://jsonplaceholder.typicode.com/todos/1’ ;

Posts: any;

//declare getposts()

getposts(){

//It should return a observable of the response from the API

this.posts = this.http.get(this.ROOT\_URL + ‘/posts’)

}

Constructor(private http: HttpClient) {}

}



Step 3:- Add HTML part

File name:- app.component.html

// button that will trigger the API call

<button (click)=”getPosts()”></button>

<div \*ngFor = “let post of posts | async”>

{{ post | json }}

</div>

Step 4:- Will create a post file

File name:- post.ts

//Now going to export interface post and fetch some properties below

export interface Post {

Id: number;

userId: number;

title: string;

body: string;

}

Step 5:- Import Post and rxjs observable

File name:- app.component.ts

Import { Post } from ‘./post’ ;

Import { Observable } from ‘rxjs/Observable’;

Export class AppComponent {

Readonly ROOT\_URL = ‘https://jsonplaceholder.typicode.com/todos/1’ ;

Posts: Observable<Post[]>;

Constructor(private http: HttpClient) {}

getposts(){

this.posts = this.http.get<Post[](this.ROOT\_URL + ‘/posts’)

}

}

Or

Export class AppComponent {

Readonly ROOT\_URL = ‘https://jsonplaceholder.typicode.com/todos/1’ ;

Posts: Observable<any>;

Constructor(private http: HttpClient) {}

getposts(){

this.posts = this.http.get<Post[](this.ROOT\_URL + ‘/posts?userID=1’)

}

**Or we can dynamically create using params**

Import { HttpClient, HttpParams } from ‘@angular/common/http’;

Export class AppComponent {

Readonly ROOT\_URL = ‘https://jsonplaceholder.typicode.com/todos/1’ ;

Posts: Observable<any>;

Constructor(private http: HttpClient) {}

getposts(){

let params = new HttpParams().set(‘userId’, ‘1’);

this.posts = this.http.get(this.ROOT\_URL + ‘/posts’, { params })

}

Step 6:- Authanticate your App

We will use HttpHeaders if we will authenticate our app

File name:- app.component.ts

Import { HttpClient, HttpParams, HttpHeaders } from ‘@angular/common/http’

Export class AppComponent {

Readonly ROOT\_URL = ‘https://jsonplaceholder.typicode.com/todos/1’ ;

Posts: Observable<any>;

Constructor(private http: HttpClient) {}

getposts(){

let headers = new HttpParams().set(‘Authorization’, ‘auth-token’);

this.posts = this.http.get(this.ROOT\_URL + ‘/posts’, { headers })

}

Step 7:- Authanticate your App

We will use HttpHeaders if we will authenticate our app

File name:- app.component.ts

Import { HttpClient, HttpParams, HttpHeaders } from ‘@angular/common/http’

export class AppComponent {

Readonly ROOT\_URL = ‘https://jsonplaceholder.typicode.com/todos/1’ ;

Posts: Observable<any>;

newPost: Observable<any>;

Constructor(private http: HttpClient) {}

getposts(){

let headers = new HttpParams().set(‘Authorization’, ‘auth-token’);

this.posts = this.http.get(this.ROOT\_URL + ‘/posts’, { headers })

}

createPost() {

const data: Post = {

id: null;

userId: 23;

title: ‘My New Post’;

body: ‘Hello World!’;

}

//will send this object as 2nd argument in the post request

This.newPost = this.http.post(this.Root\_URL + ‘/posts’, data)

}

}

HTML Part:--------

<button (click)=”createPost()”> Create Post</button>

<hr>

{{ newPost | async | json }}

**We can use rxjs to handle errors**

**File name:- app.component.ts**

Import ‘rxjs/add/operator/catch’;

Import ‘rxjs/add/operator/error’;

Import ‘rxjs/add/operator/of’;

export class AppComponent {

Readonly ROOT\_URL = ‘https://jsonplaceholder.typicode.com/todos/1’ ;

Posts: Observable<any>;

newPost: Observable<any>;

Constructor(private http: HttpClient) {}

getposts(){

let headers = new HttpParams().set(‘Authorization’, ‘auth-token’);

this.posts = this.http.get(this.ROOT\_URL + ‘/posts’, { headers })

}

createPost() {

const data: Post = {

id: null;

userId: 23;

title: ‘My New Post’;

body: ‘Hello World!’;

}

//will send this object as 2nd argument in the post request

this.newPost = this.http.post(this.Root\_URL + ‘/posts’, data)

.retry(3)

.catch(err => {

Console.log(err)

Return Observable.of(err)

})

}

}