app.UseCors(x => x.AllowAnyHeader()

.AllowAnyMethod()

.AllowAnyOrigin());

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

import { Note} from   './models/note';

import { HttpClient, HttpErrorResponse, HttpHeaders, throwError } from '@angular/common/http';

import { catchError } from 'rxjs/operators';

@Injectable({

  providedIn: 'root'

})

export class NoteService {

  errorMessage : string="";

  constructor(private \_http: HttpClient) {

  }

  notes : any[] = [];

  url : string="http://localhost:5212/api/student";

  getNotes()

  {

    const httpOptions = {

      headers: new HttpHeaders({

        'Access-Control-Allow-Origin':'\*',

        'Access-Control-Allow-Method':'\*',

      })

    };

    return this.\_http.get(this.url ,{

      headers: new HttpHeaders({

        'Content-Type': 'application/json',

        'Accept': 'application/json',

        'mode':'no-cors'

      })

    })

}

  postNote(note : Note)

  {

    return this.\_http.post(this.url,

      JSON.stringify(note));

      // httpheaders=

      // {

      //   'Accept':"application/json"

      // })

  }

}

Exception Handling

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

import { Note} from   './models/note';

import { HttpClient, HttpErrorResponse, HttpHeaders,   } from '@angular/common/http';

import { catchError } from 'rxjs/operators';

import { Observable , of, throwError} from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class NoteService {

  errorMessage : string="";

  constructor(private \_http: HttpClient) {

  }

  errorMsg : any;

  notes : any[] = [];

  url : string="http://localhost:5212/api/studen";

  getNotes()

  {

    const httpOptions = {

      headers: new HttpHeaders({

        'Access-Control-Allow-Origin':'\*',

        'Access-Control-Allow-Method':'\*',

      })

    };

    return this.\_http.get(this.url)

    .pipe(

      catchError(error => {

        let errorMsg: string;

        if (error.error instanceof ErrorEvent) {

            this.errorMsg = `Error: ${error.error.message}`;

        } else {

            this.errorMsg = this.getServerErrorMessage(error);

        }

        return throwError(this.errorMsg);

    })

);

    // .pipe(catchError(error=> this.catchAuthError(error)),);

    // return this.\_http.get(this.url ,{

    //   headers: new HttpHeaders({

    //     'Content-Type': 'application/json',

    //     'Accept': 'application/json',

    //     'mode':'no-cors'

    //   })}).pipe(catchError(error=> this.catchAuthError(error)),);

}

  postNote(note : Note)

  {

    return this.\_http.post(this.url,

      JSON.stringify(note));

      // httpheaders=

      // {

      //   'Accept':"application/json"

      // })

  }

  private getServerErrorMessage(error: HttpErrorResponse): string {

    console.log("RRRRRR " + error.status)

    switch (error.status) {

        case 404: {

            return `Not Found: ${error.message}`;

        }

        case 403: {

            return `Access Denied: ${error.message}`;

        }

        case 500: {

            return `Internal Server Error: ${error.error}`;

        }

        default: {

            return `Unknown Server Error: ${error.message}`;

        }

    }

}

}

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

import { NoteService } from '../note.service';

import { catchError } from 'rxjs';

@Component({

  selector: 'app-note',

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

  styleUrls: ['./note.component.css']

})

export class NoteComponent {

  constructor(private \_note : NoteService)

  {}

  ngOnInit()

  {

     this.\_note.getNotes().subscribe((res)=>

     {

      console.log("1111111111111222222222" + res),

     (err:any)=> console.log("1111111111111111" + err)});

}

}

<https://indepth.dev/tutorials/angular/angular-handle-http-errors-using-interceptors>

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

import { Note} from   './models/note';

import { HttpClient, HttpErrorResponse, HttpHeaders,   } from '@angular/common/http';

import { catchError } from 'rxjs/operators';

import { Observable , of, throwError} from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class Note1Service {

  errorMessage : string="";

  constructor(private \_http: HttpClient) {

  }

  errorMsg : any;

  notes : any[] = [];

  url : string="http://localhost:5212/api/studen";

  getNotes()

  {

    const httpOptions = {

      headers: new HttpHeaders({

        'Access-Control-Allow-Origin':'\*',

        'Access-Control-Allow-Method':'\*',

      })

    };

    return this.\_http.get(this.url);

  }

}

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

import {

  HttpRequest,

  HttpHandler,

  HttpEvent,

  HttpInterceptor,

  HttpErrorResponse

} from '@angular/common/http';

import { Observable, catchError, map, throwError } from 'rxjs';

@Injectable()

export class ErrorCatchingInterceptor implements HttpInterceptor {

  constructor() {}

  intercept(request: HttpRequest<unknown>, next: HttpHandler): Observable<HttpEvent<unknown>> {

    console.log("Passed through the interceptor in request");

    return next.handle(request).pipe(

            map(res => {

               console.log("Passed through the interceptor in response");

               return res

            }),

            catchError((error: HttpErrorResponse) => {

               let errorMsg = '';

               if (error.error instanceof ErrorEvent) {

                  console.log('This is client side error');

                  errorMsg = `Error: ${error.error.message}`;

               } else {

                  console.log('This is server side error');

                  errorMsg = `Error Code: ${error.status},  Message: ${error.message}`;

               }

               console.log(errorMsg);

               return throwError(errorMsg);

            })

      )

}

//   private getServerErrorMessage(error: HttpErrorResponse): string {

//     console.log("RRRRRR " + error.status)

//     switch (error.status) {

//         case 404: {

//             return `Not Found: ${error.message}`;

//         }

//         case 403: {

//             return `Access Denied: ${error.message}`;

//         }

//         case 500: {

//             return `Internal Server Error: ${error.error}`;

//         }

//         default: {

//             return `Unknown Server Error: ${error.message}`;

//         }

//     }

// }

}

 providers: [

    {

       provide: HTTP\_INTERCEPTORS,

       useClass: ErrorCatchingInterceptor,

       multi: true

    }

  ],