Steps for ErrorHandling in Angular.

Step 1: open any Angular app which Consumes service or webAPI.

Step 2 : service.ts (in out example user.service.ts)

Add the below method to handle the error

User.services.ts

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

import { Observable, throwError } from 'rxjs';

add the below method in userService class

private handleError(error:HttpErrorResponse)

    {

      if(error.status ===0)

      {

        //client side or network error occured.handle it accordingly

        console.error('An error occured',error.error);

      }

      else{

        // the backend returned an unsuccessful response code.

        // the response body may contain a clues as to went wrong.

        console.error(`Backend returned code ${error.status}  body was`,error.error);

      }

      return throwError('Something Bad happened .Please try again later');

    }

public getUsers():Observable<User []>

    {

      return this.httpclient.get<User[]>(this.dataURL)

      .pipe((catchError(this.handleError)));

    }

User.service.ts

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

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

import { Observable, throwError } from 'rxjs';

import { catchError, retry } from 'rxjs/operators';

import {User} from './Models/User';

@Injectable({

  providedIn: 'root'

})

export class UserService {

  dataURL:string='https://jsonplaceholder.typicode.com/users1';

   errorMessage:string='';

  constructor(private httpclient:HttpClient ) {

   }

    public getUsers():Observable<User []>

    {

      return this.httpclient.get<User[]>(this.dataURL)

      .pipe((catchError(this.handleError)));

    }

    private handleError(error:HttpErrorResponse)

    {

      if(error.status ===0)

      {

        this.errorMessage=`An error occured ${error.error}`;

        //client side or network error occured.handle it accordingly

        console.error('An error occured',error.error);

      }

      else{

        // the backend returned an unsuccessful response code.

        // the response body may contain a clues as to went wrong.

        this.errorMessage= `Backend returned code ${error.status}  body was ${error.error}`;

        console.error(`Backend returned code ${error.status}  body was`,error.error);

      }

      this.errorMessage +='\nSomething Bad happened .Please try again later';

      return throwError(this.errorMessage);

     // return throwError('Something Bad happened .Please try again later');

    }

}

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

import { User } from '../Models/User';

import { UserService } from '../user.service';

@Component({

  selector: 'app-user-list',

  templateUrl: './user-list.component.html',

  styleUrls: ['./user-list.component.css']

})

export class UserListComponent implements OnInit {

users:User[]=[];

ErrorMessage:string | undefined=undefined;

  constructor(private \_userservice:UserService) { }

  ngOnInit(): void {

  // this.\_userservice.getUsers().subscribe((data:User[]) =>    this.users=data);

  }

  public GetUsers()

  {

    this.\_userservice.getUsers().subscribe((data:User[]) => {this.users=data;},

    (error) => {

      this.ErrorMessage=error;

      console.log(error);

    }

    );

  }

}

<p>user-list works!</p>

<button class="btn btn-success" (click)="GetUsers()"> Get Users</button>

<section \*ngIf=ErrorMessage>

    <div class="container">

        <div class="row">

            <div class="col">

                <small class="text-danger">{{ErrorMessage}}</small>

            </div>

        </div>

    </div>

</section>

<section \*ngIf=!ErrorMessage>

    <div class="container">

        <div class="row">

            <div class="col">

<table class="table table-striped table-light">

    <thead class="thead-light">

        <th>id</th>

        <th> name</th>

        <th>Username</th>

        <th>Email</th>

    </thead>

    <tbody>

        <tr \*ngFor="let u of users">

            <td> {{u.id}}</td>

            <td> {{u.name}}</td>

            <td>{{u.username}}</td>

            <td>{{u.email}}</td>

        </tr>

    </tbody>

</table>

</div>

</div>

</div>

</section>