callback function in typescript example

In TypeScript, callback functions are commonly used to handle asynchronous operations, such as fetching data from a server, reading files, or executing tasks that take some time to complete. Here's an example of how to define and use a callback function in TypeScript:

**data.service.ts:**

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

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

import { Observable } from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class DataService {

private apiUrl = 'https://jsonplaceholder.typicode.com/posts'; // Example API URL, replace with your API endpoint

constructor(private http: HttpClient) { }

// Fetch product item data from a fake API asynchronously

fetchDataFromAPI(): Observable<ProductItem[]> {

return this.http.get<ProductItem[]>(this.apiUrl);

}

}

export interface ProductItem {

userId: number;

id: number;

title: string;

body: string;

}

**app.component.ts:**

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

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

@Component({

selector: 'app-root',

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

styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit {

fetchedData: ProductItem[];

constructor(private dataService: DataService) {}

ngOnInit() {

// Call the fetchDataFromAPI function from the DataService and subscribe to the Observable

this.dataService.fetchDataFromAPI().subscribe(

(data: ProductItem[]) => {

// Handle the fetched data

this.fetchedData = data;

},

(error) => {

console.error('Error fetching data:', error);

}

);

}

}

**app.component.html:**

<div>

<h1>Fetched Products:</h1>

<ul>

<li \*ngFor="let item of fetchedData">

<strong>Title:</strong> {{ item.title }}<br>

<strong>Body:</strong> {{ item.body }}

</li>

</ul>

</div>

**what is promises in Typescript with code examples Angular?**

In Angular applications, promises are often used to handle asynchronous operations such as making HTTP requests to fetch data from a server. Angular's HttpClient module returns Observables by default for handling HTTP requests, but you can convert these Observables into promises using the **.toPromise()** method if needed. Here's an example of using promises in Angular:

A screenshot of a video

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated A screen shot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated A screen shot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated A screen shot of a computer

Description automatically generated A computer screen shot of a check mark

Description automatically generated