Lets begin with Create a class to handle file operations (read, write, update).

Let's start by creating a FileHandler class in TypeScript that can perform read, write, and update operations on a file. We'll use Node.js' built-in fs module for file handling. Here's the code:

**Steps:**

1. Create a FileHandler class.
2. Implement methods for:
   * Writing to a file.
   * Reading from a file.
   * Updating a file.

**fileHandler.ts**

typescript

CopyEdit

import \* as fs from 'fs';

class FileHandler {

private filePath: string;

constructor(filePath: string) {

this.filePath = filePath;

}

// Write data to file (overwrites if file exists)

writeFile(data: string): void {

fs.writeFileSync(this.filePath, data, 'utf8');

console.log(`File written successfully: ${this.filePath}`);

}

// Read data from file

readFile(): string {

if (!fs.existsSync(this.filePath)) {

console.error('File does not exist.');

return '';

}

return fs.readFileSync(this.filePath, 'utf8');

}

// Append new data to file

updateFile(data: string): void {

if (!fs.existsSync(this.filePath)) {

console.error('File does not exist. Creating a new one.');

this.writeFile(data);

} else {

fs.appendFileSync(this.filePath, '\n' + data, 'utf8');

console.log(`File updated successfully: ${this.filePath}`);

}

}

}

// Example usage

const fileHandler = new FileHandler('example.txt');

fileHandler.writeFile('Hello, World!');

console.log('Read:', fileHandler.readFile());

fileHandler.updateFile('Appending new data.');

console.log('Updated Read:', fileHandler.readFile());

**Explanation:**

* **writeFile(data: string)** → Writes data to the file, overwriting existing content.
* **readFile()** → Reads and returns the file content.
* **updateFile(data: string)** → Appends data to the file if it exists, otherwise, it creates a new file.

Would you like to run this as a script and test it out? 🚀

4o

**You said:**

Yes

**ChatGPT said:**

Great! Follow these steps to run the script in Node.js:

**1️⃣ Initialize a Node.js Project (if needed)**

If you haven't already set up a Node.js project, run:

sh

CopyEdit

mkdir typescript-file-handler

cd typescript-file-handler

npm init -y

**2️⃣ Install TypeScript and Set Up Configuration**

Since we're using TypeScript, install it along with the @types/node package:

sh

CopyEdit

npm install typescript @types/node --save-dev

Then, generate a TypeScript configuration file:

sh

CopyEdit

npx tsc --init

**3️⃣ Create fileHandler.ts**

Save the following code in fileHandler.ts:

typescript

CopyEdit

import \* as fs from 'fs';

class FileHandler {

private filePath: string;

constructor(filePath: string) {

this.filePath = filePath;

}

// Write data to file (overwrites existing content)

writeFile(data: string): void {

fs.writeFileSync(this.filePath, data, 'utf8');

console.log(`File written successfully: ${this.filePath}`);

}

// Read data from file

readFile(): string {

if (!fs.existsSync(this.filePath)) {

console.error('File does not exist.');

return '';

}

return fs.readFileSync(this.filePath, 'utf8');

}

// Append new data to file

updateFile(data: string): void {

if (!fs.existsSync(this.filePath)) {

console.error('File does not exist. Creating a new one.');

this.writeFile(data);

} else {

fs.appendFileSync(this.filePath, '\n' + data, 'utf8');

console.log(`File updated successfully: ${this.filePath}`);

}

}

}

// Example usage

const fileHandler = new FileHandler('example.txt');

fileHandler.writeFile('Hello, World!');

console.log('Read:', fileHandler.readFile());

fileHandler.updateFile('Appending new data.');

console.log('Updated Read:', fileHandler.readFile());

**4️⃣ Compile TypeScript to JavaScript**

Run:

sh

CopyEdit

npx tsc fileHandler.ts

This generates a JavaScript file fileHandler.js.

**5️⃣ Run the Script**

Execute the compiled JavaScript file:

sh

CopyEdit

node fileHandler.js

You should see output similar to:

yaml

CopyEdit

File written successfully: example.txt

Read: Hello, World!

File updated successfully: example.txt

Updated Read: Hello, World!

Appending new data.

Check the example.txt file—it should contain:

cpp

CopyEdit

Hello, World!

Appending new data.