What is the purpose of the Node.js 'fs' module?

Topics : Node js
Written on November 30, 2023

The `fs` (file system) module in Node.js provides an interface for interacting with the file system. It allows you to perform various file-related operations, such as reading from or writing to files, creating directories, and manipulating file metadata.

Here are some commonly used methods in the `fs` module:

Reading a File:

- `fs.readFile(path[, options], callback)`: Reads the entire contents of a file asynchronously.

```
const fs = require('fs');

fs.readFile('example.txt', 'utf8', (err, data) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log(data);
});
```

Writing to a File:

- `fs.writeFile(file, data[, options], callback)`: Writes data to a file asynchronously, replacing the file if it already exists.

```
const fs = require('fs');

fs.writeFile('example.txt', 'Hello, Node.js!', 'utf8', (err) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log('File written successfully');
});
```

Reading a Directory:

- `fs.readdir(path[, options], callback)`: Reads the contents of a directory asynchronously.

```
const fs = require('fs');

fs.readdir('.', (err, files) => {
    if (err) {
        console.error(err);
        return;
    }
    console.log('Files in the current directory:', files);
});
```

Creating a Directory:

- `fs.mkdir(path[, options], callback)`: Creates a new directory asynchronously.

```
const fs = require('fs');

fs.mkdir('newDirectory', (err) => {
    if (err) {
        console.error(err);
        return;
    }
    console.log('Directory created successfully');
});
```

Checking if a File or Directory Exists:

- `fs.exists(path, callback)`: Checks if a file or directory exists.

```
const fs = require('fs');

fs.exists('example.txt', (exists) => {
    console.log(`File exists: ${exists}`);
});
```

File Stats:

- `fs.stat(path, callback)`: Gets the file status asynchronously, providing information like size, modification time, and more.

```
const fs = require('fs');

fs.stat('example.txt', (err, stats) => {
```

```
    if (err) {
        console.error(err);
        return;
    }
    console.log('File size:', stats.size);
    console.log('Last modified:', stats.mtime);
});
```

Deleting a File:

- `fs.unlink(path, callback)`: Deletes a file asynchronously.

```
const fs = require('fs');

fs.unlink('example.txt', (err) => {
    if (err) {
        console.error(err);
        return;
    }
    console.log('File deleted successfully');
});
```