BÀI BÁO CÁO   
TRUY VẤN TỆP TIN

**\*Kiến thức chung:** Công cụ: Var fs = require(‘fs’) ;  
Mọi phương thức trong mô-đun đều có 2 dạng: Đồng bộ và không đồng bộ

-Đồng bộ **(synchronous)**: code sẽ chạy tuần tự theo trình tự đã viết sẵn

-Không đồng bộ **(asynchronous)**: code ở dưới có thể tiếp tục chạy mặc dù đoạn code ở trên chưa được xử lí hết và trả về kết quả.

🡪 Tốt hơn là sử dụng một phương thức không đồng bộ thay vì một phương thức đồng bộ, vì phương thức thứ nhất không bao giờ chặn một chương trình trong quá trình thực thi, trong khi phương thức thứ hai thì có.

1. **Cú pháp đọc tệp**

Các fs.readFile() phương pháp được sử dụng để đọc các tập tin trên máy tính của bạn.

Ví dụ:

Ta có file :

filedemo.html

<html>  
<body>  
<h1>My Header</h1>  
<p>My paragraph.</p>  
</body>  
</html>

Đọc tệp trên:

var http = require('http');  
var fs = require('fs');  
http.createServer(function (req, res) {  
**fs.readFile('filedemo.html', function(err, data) {**    res.writeHead(200, {'Content-Type': 'text/html'});  
    res.write(data);  
    return res.end();  
  });  
}).listen(8080);

1. **Cú pháp tạo tệp**

Mô-đun Hệ thống Tệp có các phương pháp để tạo tệp mới như:

* fs.appendFile()
* fs.open()
* fs.writeFile()

Các fs.appendFile()phương pháp gắn thêm nội dung cụ thể vào một tập tin. Nếu tệp không tồn tại, tệp sẽ được tạo

Ví dụ:

var fs = require('fs');  
  
fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {  
  if (err) throw err;  
  console.log('Saved!');  
});

Các fs.open()phương pháp có một "flag" như là đối số thứ hai, nếu lá cờ là "w" cho "writing", các tập tin định được mở ra cho văn bản. Nếu tệp không tồn tại, một tệp trống sẽ được tạo:

Ví dụ:

var fs = require('fs');  
  
fs.open('mynewfile2.txt', 'w', function (err, file) {  
  if (err) throw err;  
  console.log('Saved!');  
});

Các fs.writeFile()phương pháp thay thế các tập tin và nội dung cụ thể nếu nó tồn tại. Nếu tệp không tồn tại, một tệp mới, chứa nội dung được chỉ định, sẽ được tạo:

Ví dụ:

var fs = require('fs');  
  
fs.writeFile('mynewfile3.txt', 'Hello content!', function (err) {  
  if (err) throw err;  
  console.log('Saved!');  
});

1. **Cập nhật tệp**

Mô-đun Hệ thống Tệp có các phương pháp để cập nhật tệp:

* fs.appendFile()
* fs.writeFile()

Các fs.appendFile()phương pháp gắn thêm các nội dung quy định tại phần cuối của tập tin nhất định:

Ví dụ: // Nối “Văn bản của tôi” vào “mynew” //

var fs = require('fs');  
  
fs.appendFile('mynew.txt', ' Văn bản của tôi', function (err) {  
  if (err) throw err;  
  console.log('Updated!');  
});

Các fs.writeFile()phương pháp thay thế các tập tin và nội dung cụ thể:

Ví dụ: // Thay thế nội dung tệp “mynew.txt” //

var fs = require('fs');  
  
fs.writeFile('mynew.txt', 'This is my text', function (err) {  
  if (err) throw err;  
  console.log('Replaced!');  
});

1. **Xóa tệp tin**

Để xóa tệp bằng mô-đun Hệ thống tệp, hãy sử dụng fs.unlink() phương pháp.

Các fs.unlink()phương pháp xóa các tập tin nhất định:

Ví dụ: // Xóa file “mynew.txt” //

var fs = require('fs');  
  
fs.unlink('mynewfile2.txt', function (err) {  
  if (err) throw err;  
  console.log('File deleted!');  
});

1. **Đổi tên tệp**

Để đổi tên tệp bằng mô-đun Hệ thống tệp, hãy sử dụng fs.rename() phương pháp này.

Các fs.rename()phương pháp đổi tên các tập tin nhất định:

Ví dụ: // Đổi tên “myname” thành “myrename”

var fs = require('fs');  
  
fs.rename('myname.txt', 'myrenamed.txt', function (err) {  
  if (err) throw err;  
  console.log('File Renamed!');  
});

1. **Cú pháp viết tệp**

Sau đây là cú pháp của một trong các phương pháp để ghi vào tệp: fs.writeFile (tên tệp, dữ liệu [, tùy chọn], gọi lại)   
Phương thức này sẽ ghi đè tệp nếu tệp đã tồn tại. Nếu bạn muốn ghi vào một tệp hiện có, thì bạn nên sử dụng một phương pháp khác có sẵn. Đây là mô tả của các tham số được sử dụng:   
  
• path - Đây là chuỗi có tên tệp bao gồm cả đường dẫn.   
• dữ liệu - Đây là Chuỗi hoặc Bộ đệm được ghi vào tệp.   
• options - Tham số thứ ba là một đối tượng sẽ chứa {encoding, mode, flag}. Theo mặc định, mã hóa là utf8, chế độ là giá trị bát phân 0666 và flag là 'w'   
• callback - Đây là hàm gọi lại nhận một lỗi tham số duy nhất trả về lỗi trong trường hợp có bất kỳ lỗi viết nào