NodeJS là một ngôn ngữ giúp bạn xử lý hàng triệu request từ phía người dùng. Vấn đề xử lý đó cực kỳ quan trọng nếu bạn xử dụng các ngôn ngữ phổ thông như PHP hoặc ASP.NET thì rất có thể xảy ra tình trạng nghẽn Server làm cho hệ thống hoàn toàn bị tê liệt.

Vậy nếu trường hợp tôi có một Website xem phim hoặc nghe nhạc thì vấn đề của lập trình viên không còn là code nữa mà là về hệ thống. Dù cho bạn có tốt và tối ưu tới đâu đi nữa nếu code đó không hỗ trợ Server giảm tải thì trước sau Website của bạn cũng bị tê liệt vì quá tải. Nên ở trong phần này tôi sẽ hướng dẫn các bạn các chúng ta giảm tải cho Website đặc biệt là Website xem phim bằng cách Streaming video để tránh trường hợp video load đồng loạt cùng một thời điểm.

* ReadStream File

var fs = require(‘fs’);

var readStream = fs.createReadStream(‘index.txt’);

var content = ‘’;

Cách 1:

readStream.on(‘data’, function(chunk){ // Hàm đọc file, mỗi lần đọc cắt từng khúc(chunk) để đọc

content += chunk;

});

readStream.on(‘end’, function(){ // Hàm kết thúc khi đọc xong file

console.log(content)

});

Cách 2:

var chunk;

readStream.on(‘readable’, function(){

while((chunk = readStream.read()) != null){

content += chunk;

}

})

readStream.on(‘end’, function(){

console.log(content)

});

* WriteStream File

var fs = require(‘fs’);

var writeStream = fs.createWriteStream(‘data.txt’);

writeStream.write(‘Dữ liệu cần ghi’);

writeStream.end(); // Kết thúc ghi dữ liệu

* PIPE: là luồng dữ liệu chuyển từ chỗ này qua chỗ khác thông qua 1 thằng khác xử lý

var fs = require(‘fs’)

var readStream = fs.createReadStream(‘data.txt’);

var writeStream = fs.createWriteStream(‘save.txt’);

readStream.pipe(writeStream) // đọc dữ liệu từ readStream rồi copy sang writeStream

* Nén file bằng Zlib (Zip Lirary)

var zlib =require(‘zlib’);

var fs = require(‘fs’);

var rS = fs.createReadStream(‘data.txt’);

var wS = fs.createWriteStream(‘file.txt.zip’);

rS.pipe(zlip.createGzip()).pipe(wS).on(‘finish’, function(){

console.log(‘Success’);

})

//rS đọc dữ liệu nén lại rồi chuyển qua wS ghi file.

* Duplex (Read - Write)

var Duplex = require(‘stream’).Duplex;

var fs = require(‘fs’);

Duplex.Readable = fs.createReadStream(‘data.txt’);

Duplex.WriteAble = fs.createWriteStream(‘save.txt’);

Duplex.Readable.on(‘data’, function(chunk){

Duplex.Writeable.write(‘chunk’);

});

Duplex.Readable.on(‘end’, function(){

console.log(‘Complete’)

})