# BÁO CÁO HỌC NODEJS

# Tổng quan về Node.js

* Node.js là một Framework mạnh mẽ dựa trên nền tảng Google Chrome Javascript V8 Engine.
* Node.js được dùng để phát triển các ứng dụng hướng nhiều đến I/O trên nền tảng web như Video Streaming Site, các ứng dụng Single-page App hay các ứng dụng web khác.
* NodeJs hoàn toàn mã nguồn mở, miễn phí và được sử dụng rất rộng rãi phổ biến hiện nay.

# Tại sao lại là Node.js

* Đây là cách PHP hoặc ASP xử lý một yêu cầu tệp:
  1. Gửi tác vụ đến hệ thống tệp của máy tính.
  2. Chờ trong khi hệ thống tệp mở và đọc tệp.
  3. Trả về nội dung cho khách hàng.
  4. Sẵn sàng để xử lý các yêu cầu tiếp theo.
* Đây là cách Node.js xử lý yêu cầu tệp:
  1. Gửi tác vụ đến hệ thống tệp của máy tính.
  2. Sẵn sàng để xử lý các yêu cầu tiếp theo.
  3. Khi hệ thống tệp đã mở và đọc tệp, máy chủ sẽ trả lại nội dung cho máy khách.
* Node.js bỏ phần chờ đợi và tiếp tục bước tiếp theo, nên tốc độ nhanh hơn và hiệu quả về bộ nhớ.

# Module

* Module như kiểu là một tập hợp các chức năng ( giống như là một Function ) mà bạn muốn đưa vào chương trình.
* Node.js có một bộ Module mà bạn có thể sử dụng mà không cần cài đặt thêm.
* Cách sử dụng Module:
  + Sử dụng Require() với tên của Module cần gọi.
  + VD: var http = require('http'); bây giờ có thể truy cập vào Module HTTP và có thể tạo máy chủ.
  + Chúng ta có thể tạo được Module của riêng mình rồi gọi Module theo cú pháp như trên:
  + VD: Tên tệp js là mymodule.js sau đó viết các code bạn cần giống kiểu 1 chức năng sau đó bạn sẽ gọi nó ra để sử dụng

# Module HTTP

* Node.js có một mô-đun tích hợp có tên HTTP, cho phép Node.js truyền dữ liệu qua Giao thức truyền văn bản Hyper (HTTP).
* Để sử dụng Module HTTP sử dụng phương thức require():

Var http = require(‘http’);

* Mô-đun HTTP có thể tạo một máy chủ HTTP lắng nghe các cổng máy chủ và trả lời lại cho máy khách.
* VD ta sử dụng phương thức createServer() để tạo máy chủ http:



* Máy chủ HTTP đã được tạo khi bạn khởi chạy cổng 8080 trên trình duyệt sẽ hoạt động và in ra phản hồi phía máy khách là dòng chữ Hello World!
* Hàm được truyền vào http.createServer() có một reqđối số thể hiện yêu cầu từ máy khách, dưới dạng một đối tượng (http.IncomingMessage object).
* Đối tượng này có thuộc tính gọi là “url” chứa 1 phần sau tên miền nó là 1 kiểu gửi yêu cầu từ máy khách lên máy chủ thực hiện hoặc nhận thông tin.

# Module File System

* Module File System cho phép ta thao tác với các tệp trên máy tính của mình.
* Để sử dụng Module này ta vẫn sử dụng phương thức require(): Var fs = require(‘fs’);
* Module File System được sử dụng phổ biến để:

1. Đọc File
2. Tạo File
3. Update File
4. Xóa File
5. Đổi tên File

* Để đọc File ta sử dụng phương thức: fs.readFile()
* Tương tự các phương thức để tạo File:

1. fs.appendFile()
2. fs.open()
3. fs.writeFile()

* Cập nhật File sử dụng :

1. fs.appendFile()
2. fs.writeFile()

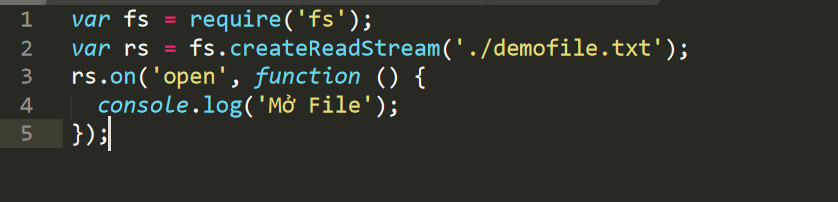
* Xóa File sử dụng: fs.unlink()
* Đổi tên File sử dụng: fs.rename()

# Node.js NPM

* NPM là trình quản lý gói cho các gói Node.js hoặc Module nếu muốn.
* Chương trình NPM được cài đặt trên máy tính khi cài đặt Node.js.
* Có thể sử dụng NPM để cài các gói với CMD: npm install upper-case

# Sự kiện trong Node.js

* Các đối tượng trong Node.js có thể kích hoạt các sự kiện, như đối tượng readStream kích hoạt các sự kiện khi mở và đóng tệp.



* Để sử dụng Module Sự kiện, sử dụng phương thức require() . Ngoài ra, tất cả các thuộc tính và phương thức sự kiện là một thể hiện của một đối tượng EventEuctor. Để có thể truy cập các thuộc tính và phương thức này, hãy tạo một đối tượng EventEuctor:

var events = require('events');

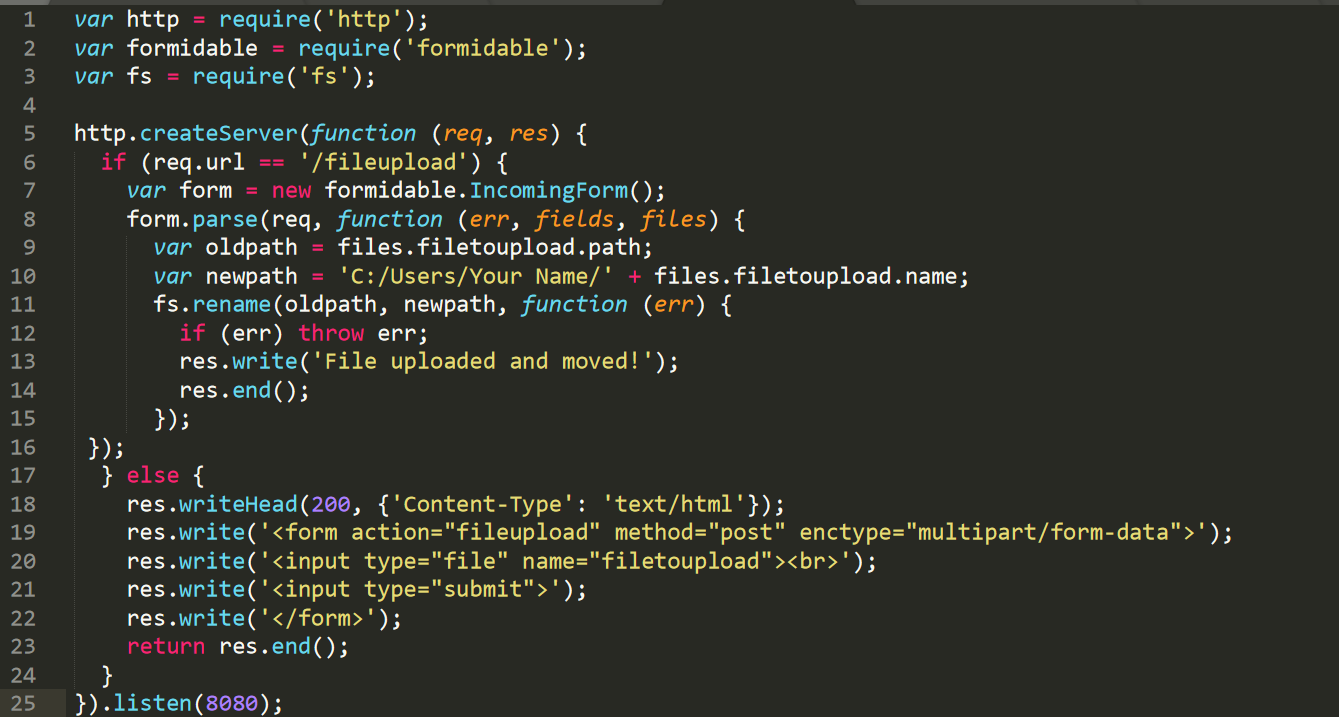
var eventEmitter = new events.EventEmitter();

# Upload Files trong Node.js

* Để sử dụng Upload Files trong Node.js ta sử dụng Module Formidable.
* Cụ thể là gọi: var formidable = require('formidable');



* Khi một tập tin được tải lên thành công vào máy chủ, nó được đặt trên một thư mục tạm thời.
* Đường dẫn đến thư mục này có thể được tìm thấy trong đối tượng "files", được truyền dưới dạng đối số thứ ba trong parse().
* Để di chuyển tệp vào thư mục bạn chọn, hãy sử dụng Module File System và đổi tên tệp:



# Gửi Email trong Node.js

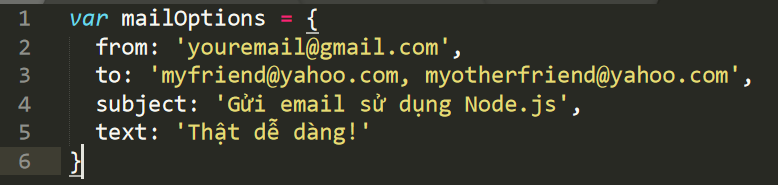
* Bằng cách sử dụng Module Nodemailer.
* Module Nodemailer có thể được tải xuống và cài đặt bằng npm:

npm install nodemailer

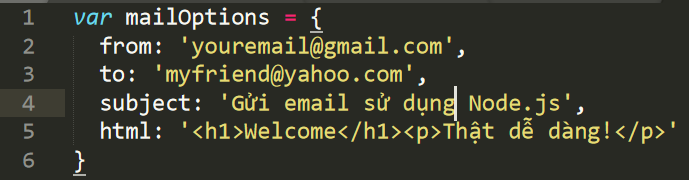
* Chúng ta vẫn sử dụng như bình thường: var nodemailer = require('nodemailer');



* Sử dụng tên người dùng và mật khẩu từ nhà cung cấp email đã chọn của bạn để gửi email.
* Để gửi email đến nhiều người nhận, hãy thêm chúng vào thuộc tính "to" của đối tượng mailOptions, được phân tách bằng dấu phẩy:



* Để gửi văn bản có định dạng HTML trong email của bạn, hãy sử dụng thuộc tính "html" thay vì thuộc tính "văn bản":

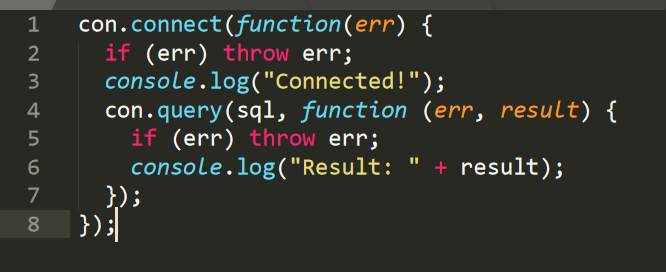


# MySql trong Node.js

* Chúng ta hoàn toàn có thể sử dụng cơ sở dữ liệu MySql để tạo và quản lý dữ liệu của mình thông qua nó.
* Trong Node.js việc sử dụng MySql rất dễ dàng với chỉ 1 cú pháp cài đặt: npm install mysql
* Để sử dụng chúng ta dùng require() như với Module: var mysql = require (‘mysql’);
* Tạo kết nối ta sử dụng tên và mật khẩu từ cơ sở dữ liệu MYSQL của mình:



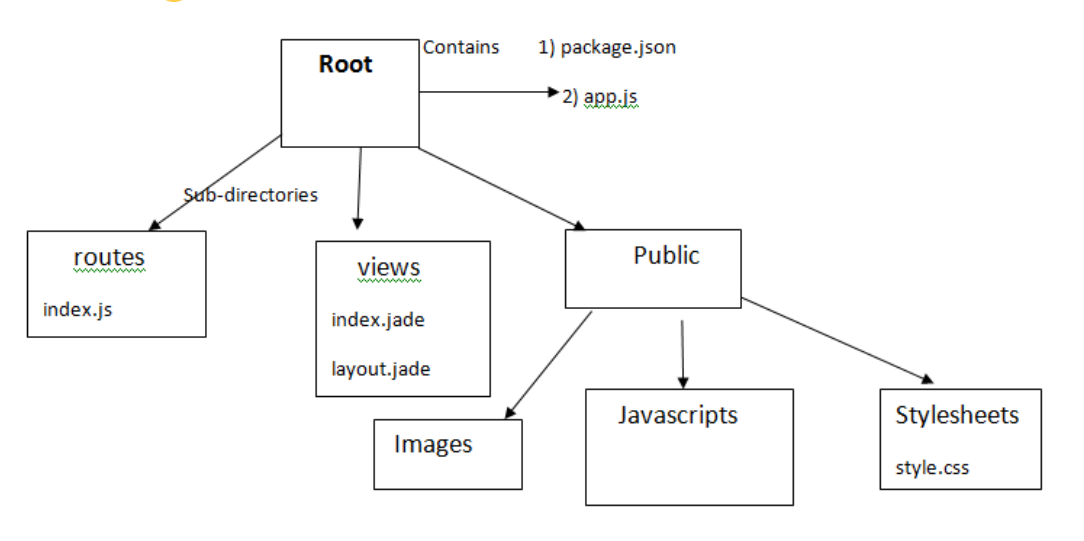
* Để truy vấn cơ sở dữ liệu ta sử dụng các câu lệnh truy vấn SQL



* Việc tạo database và các truy vấn, thao tác với database cũng rất dễ dàng.

# Sử dụng FrameWork của Node.js: Express

* Express js là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile.
* Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ nên không phải lo lắng khi làm việc với Framework này.
* Về performance: Express cung cấp thêm về các tính năng để lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.
* Và hơn hết, các Framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function, chẳng hạn: SailsJS, MEAN,....
* Để cài đặt Express ta sử dụng : npm install express
* Cấu trúc của ExpressJS:



* Cấu trúc của express js cũng dễ hiểu.

1. app.js chứa các thông tin về cấu hình, khai báo, các định nghĩa,... để ứng dụng của chúng ta chạy ok.
2. package.json chứa các package cho ứng dụng chạy. Nếu bạn nào làm với PHP hoặc RoR rồi thì file này có chức năng tương tự như composer.json hoặc Gemfile 😃
3. Folder routes: chứa các route có trong ứng dụng
4. Folder view: chứa view/template cho ứng dụng
5. Folder public chứa các file css, js, images,...cho ứng dụng.

# KẾT THÚC BÁO CÁO Ạ ! CÓ VẤN ĐỀ GÌ ANH CHỈ BẢO EM VỚI Ạ ! EM CẢM ƠN ANH Ạ !