**REPORT MEANJS**

1. THÔNG TIN NHÓM

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Phạm Trọng Ân | 1312024 |
| 2 | Nguyễn Hửu Quân | 1312461 |

1. NỘI DUNG

Để thuận tiện cho việc xây dựng, quản lý và phát triển các trang web, rất nhiều các framework đã. Nổi tiếng nhất trong số đó được xây dựng phải kể đến LAMP bao gồm Linux server, Apache, MySQL và PHP. Tuy nhiên, web hiện đại yêu cầu khả năng tương tác cao hơn, đồng nghĩa với việc LAMP stack không phải là giải pháp tối ưu. Đã có rất nhiều các framework khác được xây dựng dựa trên nền tảng các ngôn ngữ khác nhau như Ruby, Python, Perl, …. Tuy nhiên, về cơ bản vẫn dựa trên LAMP và ít có sự phá cách. Thời gian gần đây, nhiều website đã được xây dựng dựa trên NodeJS tuy nhiên vẫn còn hết sức đơn giản vì NodeJS vẫn còn hết sức thô sơ. Bài viết này xin được phép giới thiệu một framework mới dựa trên NodeJS có khả năng xây dựng một trang web hoàn chỉnh dựa trên ngôn ngữ Javascript, đó là MEANJS.

# Giới thiệu về MeanJS

MeanJS là một framework hoàn toàn mới được xây dựng dựa trên các thành phần bao gồm:

* **M**ongoDB: Một NoSQL database điển hình.
* **E**xpress: Middleware với database.
* **A**ngularJS: front end framework.
* **N**odeJS: một ứng dụng web về phía Server.

## Giới thiệu MongoDB

MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

### Khái niệm Database

Database là một nơi chứa vật lý cho các Collection. Mỗi Database lấy tập hợp các file riêng của nó trên hệ thống file. Mỗi MongoDB Server có thể có nhiều cơ sở dữ liệu.

### Khái niệm Collection

Collection là một nhóm các Document trong MongoDB. Nó tương đương như một bảng trong RDBMS. Do đó, một Collection tồn tại bên trong một cơ sở dữ liệu duy nhất. Các Collection không có ràng buộc Relationship như các hệ quản trị cơ sở dữ liệu khác nên việc truy xuất rất nhanh, chính vì thế mỗi collection có thể chứa nhiều thể loại khác nhau không giống như table trong hệ quản trị mysql là các field cố định. Các Document bên trong một Collection có thể có nhiều trường khác nhau. Đặc biệt, tất cả các Document trong một Collection là tương tự nhau hoặc với cùng mục đích liên quan.

### Khái niệm Document

Một Document trong MongoDB, có cấu trúc tương tự như kiểu dữ liệu JSON, là một tập hợp các cặp key-value. Các Document có schema động, nghĩa là Document trong cùng một Collection không cần thiết phải có cùng một tập hợp các trường hoặc cấu trúc giống nhau, và các trường chung trong Document của một Collection có thể giữ các kiểu dữ liệu khác nhau.

### Cấu trúc Document đơn giản

Ví dụ dưới đây minh họa cấu trúc Document của một Blog site với một cặp key-value phân biệt bởi dấu phẩy.



Ở đây, \_id là một số thập lục phân 12 byte để đảm bảo tính duy nhất của mỗi Document. Bạn có thể cung cấp \_id trong khi chèn vào Document. Nếu bạn không cung cấp, thì mongoDB sẽ cung cấp một \_id duy nhất cho mỗi Document. Trong 12 byte này, 4 byte đầu là cho Timestamp hiện tại, 3 byte tiếp theo cho ID của thiết bị, 2 byte tiếp là process id của MongoDB Server và 3 byte còn lại là giá trị có thể tăng.

### Cơ chế hoạt động

Mỗi bản ghi của mongodb được tự động gắn thêm một field có name là “id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi so với bản ghi khác. Trường dữ liệu “id” luôn được tự động đánh index để tốc độ truy vấn thông tin đạt hiệu xuất cao nhất. Mỗi khi có một truy vấn dữ liệu thì bản ghi được ghi đệm lên bộ nhớ RAM để lượt truy vấn sau diễn ra nhanh hơn mà không cần đọc từ ổ cứng. Khi có yêu cầu: “thêm, sửa, xóa” vào bản ghi, để đảm bảo hiệu xuất của ứng dụng, mặc định thì mongodb chưa cập nhật dữ liệu xuống ổ cứng ngay lập tức mà sau 60 giây thì mongodb mới ghi toàn bộ dữ liệu từ RAM xuống ổ cứng.

### Ưu điểm

Lưu trữ phi cấu trúc, không có tính ràng buộc toàn vẹn nên tính sẵn sàng cao, hiêu suất lớn, dễ dàng mở rộng lưu trữ. Dữ liệu được ghi đệm lên RAM hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

### Nhược điểm

Không ràng buộc toàn vẹn nên không ứng dụng được cho các mô hình giao dịch cần độ chính xác cao, không có cơ chế transaction để phục vụ các ứng dụng ngân hàng. Dữ liệu được catching từ RAM làm trọng tâm hoạt động nên khi hoạt động cần bộ nhớ RAM lớn. Khả năng mất dữ liệu do mất điện với hiệu xuất cao do dữ liệu không được ghi lập tức.

## Giới thiệu Express Framework

Express là một framework nhỏ và tiện ích để xây dựng các ứng dụng web, cung cấp một lượng lớn của tính năng mạnh mẽ để phát triển các ứng dụng web và mobile. Nó rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web. Dưới đây là các tính năng cơ bản của Express framework.

* Cho phép thiết lập các lớp trung gian để trả về các HTTP request.
* Định nghĩa bảng routing có thể được sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template.

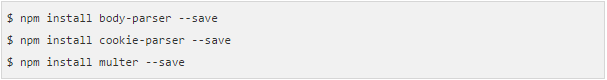
### Cài đặt Express Framework

Đầu tiên, cài đặt Express framework sử dụng npm như sau:



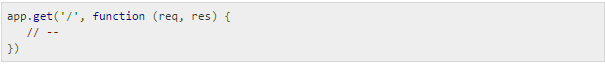
Lệnh trên lưu phần cài đặt trong mục node\_modules và tạo thư mục express bên trong thư mục đó. Dưới đây là các thành phần module quan trọng được cài đặt cùng với express:

* body-parser: Đây là một lớp trung gian node.js để xử lý JSON, dữ liệu thô, text và mã hóa URL.
* cookie-parser: Chuyển đổi header của Cookie và phân bố đến các req.cookies.
* multer: Đây là một thành phần trung gian trong node.js để xử lý phần multipart/from-data.



### Đối tượng Request và Response trong Node.js

Ứng dụng Express sử dụng một hàm callback có các tham số là các đối tượng request và response.



Bạn có thể tham khảo chi tiết về 2 đối tượng này dưới đây:

* Đối tượng Request: Đối jg này biểu diễn một HTTP request và có các thuộc tính cho các request như các chuỗi truy vấn, tham số, body, HTTP header và những phần khác.
* Đối tượng Response: Đối tượng này biểu diễn HTTP response được ứng dụng Express gửi đi khi nó nhận về một HTTP request.

Bạn có thể in ra các đối tượng req và res để cung cấp một lượng lớn thông tin liên quan đến HTTP request và trả về cookie, session, URL, ….

## Giới thiệu AngularJS

AngularJS là một framework ứng dụng web mã nguồn mở. Nó được phát triển lần đầu năm 2009 bởi Misko Hevery và Adam Abrons. Hiện tại nó được duy trì bởi Google.

Định nghĩa về AngularJS được đưa ra chính thức như sau: AngularJS là một framework có cấu trúc cho các ứng dụng web động. Nó cho phép bạn sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và súc tích. Hai tính năng cốt lõi: **Data binding** và **Dependency injection** của AngularJS loại bỏ phần lớn code mà bạn thường phải viết. Nó xảy ra trong tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào.

### Các tính năng chung của AngularJS

AngularJS là một Framework phát triển mạnh mẽ dựa trên JavaScript để tạo các ứng dụng RICH Internet Application (RIA).

AngularJS cung cấp cho lập trình viên những tùy chọn để viết các ứng dụng client-side trong mô hình MVC (Model View Controller) một cách rõ ràng.

Các ứng dụng được viết bởi AngularJS tương thích với nhiều phiên bản trình duyệt web. AngularJS tự đọng xử lý mã JavaScript để phù hợp với mỗi trình duyệt.

AngularJS có mã nguồn mở, miễn phí hoàn toàn, được sử dụng bởi hàng ngàn lập trình viên trên thế giới. Nó hoạt động dưới giấy phép Apache License version 2.0.

Nhìn chung, AngularJS là một framework để tạo các ứng dụng lớn, các hiệu năng cao trong khi giữ cho chúng có thể dễ dàng duy trì.

### Các tính năng cốt lõi của AngularJS

Dưới đây là các tính năng cốt lõi của AngularJS:

* Data-binding: Nó tự động đồng bộ hóa dữ liệu giữa thành phần model và view.
* Scope: Là những đối tượng hướng đến model, nó hoạt động như là cầu nối giữa controller và view.
* Controller: Đây là những tính năng của AngularJS mà được giới hạn tới một scope cụ thể.
* Service: AngularJS hoạt động với một vài dịch vụ (service) có sẵn, ví dụ $http để tạo XML HttpRequests. Nó là các singleton object mà được khởi tạo duy nhất một lần trong ứng dụng.
* Filter: Nó lựa chọn (hay là lọc) các tập con từ tập item trong các mảng và trả về các mảng mới.
* Directive: Directive là các marker trong các phần tử DOM (như các phần tử, thuộc tính, css và nhiều hơnthees). Nó có thể dùng để tạo các thẻ HTML riêng phục vụ những mục đích riêng. AngularJS có những directive có sẵn như ngBind, ngModel…
* Template: Là các rendered view với các thông tin từ controller và model. Nó có thể được sử dụng trong các file riêng rẽ (ví dụ như index.jsp) hoặc nhiều view với một trang sử dụng “partials”.
* Routing: Là khái niệm của sự chuyển dịch qua lại các view.
* Model View Whatever: MVC là một mô hình thiết kế để phân chi các ứng dụng thành nhiều phần khác nhau (gọi là Model, View, Controller), một phần sử dụng với một nhiệm vụ nhất định. AngularJS không triển khai MVC theo cách truyền thống, mà gắn liền hơn với Model-View-ViewModel. Nhóm phát triển AngularJS đã đặt tên vui cho mô hình này là Model View Whatever.
* Deep Linking: Cho pháp bạn mã hóa trạng thái các ứng dụng trên địa chỉ URL để nó có thể được bookmark. Các ứng dụng có thể được phục hồi lại từ các địa chỉ URL với cùng một trạng thái.
* Dependency Injection: AngularJS có sẵn một hệ thống con dependency injection để giúp các lập trình viên tạo ra các ứng dụng dễ phát triển, dễ hiểu, dễ kiểm tra.

### Ưu điểm của AngularJS

* AngularJS cung cấp khả năng tạo ra các Single Page Application một các rất rõ ràng và dễ dàng để duy trì.
* AngularJS cung cấp khả năng Data binding tới HTML do đó giúp người dùng cảm giác linh hoạt, thân thiện.
* AngularJS code dễ dàng khi unit test.
* AngularJS sử dụng denpendency injection.
* AngularJS cung cấp khả năng tái sử dụng các component (thành phần).
* Với AngularJS, lập trình viên sẽ viết ít code hơn, với nhiều chức năng hơn.
* Với AngularJS, view là thành phần trong trang HTML thuần, trong khi controller được viết bởi JavaScript với quá trình xử lý nghiệp vụ.
* Và trên tất cả, ứng dụng AngularJS có thể chạy trên hết các trình duyệt web, trên các nền tảng Android và iOS.

### Nhược điểm của AngularJS

Mặc dù AngularJS có thể kể đến rất nhiều các ưu điểm, nhưng đến thời điểm này, nó vẫn có một số điểm yếu sau:

* Không an toàn: Là một JavaScript framework, ứng dụng được viết bởi AngularJS không an toàn. Phải có các tính năng bảo mật và xác thực phía server sẽ giúp ứng dụng trở nên an toàn hơn.
* Nếu người sử dụng ứng dụng của bạn vô hiệu hóa JavaScript thì họ chỉ nhìn được trang cơ bản, không thấy gì thêm.

### Các thành phần của AngularJS

AngularJS framework có thể được chia thành ba thành phần chính sau:

* **ng-app**: directive này định nghĩa và liên kết một ứng dụng AngularJS tới HTML.
* **ng-model**: directive này gắn kết giá trị của dữ liệu ứng dụng AngularJS đến các điều khiển đầu vào HTML.
* **ng-bind**: directive này gắn kết dữ liệu ứng dụng AngularJS đến các thẻ HTML.

## Giới thiệu NodeJS

NodeJS là một nền tảng Server side được xây dựng dựa trên JavaScript Engine (V8 Engine). Node.js được phát triển bởi Ryan Dahl năm 2009. Định nghĩa NodeJS bởi tài liệu chính thức như sau: “Node.js là một nền tảng dựa vào Chrome Javascript runtime để xây dựng các ứng dụng nhanh, có độ lớn. Node.js sử dụng các phần phát sinh các sự kiện (event-driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.”

NodeJS là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía Server và các ứng dụng liên quan đến mạng. Ứng dụng Node.js được viết bằng javascript và có thể chạy trong môi trường Node.js trên hệ điều hành Windows, Linux,….

Node.js cũng cung cấp cho chúng ta các module Javascript đa dạng, có thể đơn giản hóa sự phát triển của các ứng dụng web sử dụng Node.js với các phần mở rộng.



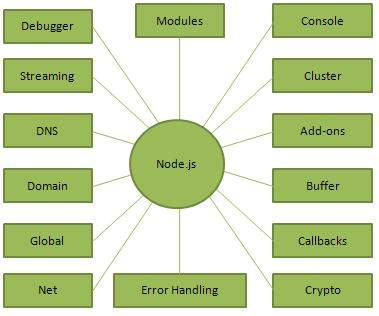
### Đặc điểm của Node.js

Dưới đây là vài đặc điểm quan trọng biến Node.js trở thành sự lựa chọn hàng đầu trong phát triển phần mềm:

* Không đồng bộ và Phát sinh sự kiện (Event Driven): Tất cả các APIs của thư viện Node.js đều không đồng bộ, nghĩa là không blocking (khóa). Nó rất cần thiết vì Node.js không bao giờ đợi một API trả về dữ liệu. Server chuyển sang một API sau khi gọi nó và có cơ chế thông báo về Sự kiện của Node.js giúp Server nhận được phản hồi từ các API gọi trước đó.
* Chạy rất nhanh: Dựa trên V8 Javascript Engine của Google Chrome, thư viện Node.js rất nhanh trong các quá trình thực hiện code.
* Các tiến trình đơn giản nhưng hiệu năng cao: Node.js sử dụng một mô hình luồng đơn (single thread) với các sự liện lặp. Các cơ chế sự kiện giúp Server trả lại các phản hồi với một cách không khóa và tạo cho Server hiệu quả cao ngược lại với các cách truyền thống tạo ra một số lượng luồn hữu hạn để quản lý request. Node.js sử dụng các chương trình đơn luồng và các chương trình này cung cấp các dịch vụ cho số lượng request nhiều hơn so với các Server truyền thống như Apache HTTP Server.
* Không đệm: Ứng dụng Node.js không lưu trữ các dữ liệu buffer.
* Có giấy phép: Node.js được phát hành dựa vào MIT License.

### Các thành phàn quan trọng trong Node.js

Lược đồ dưới đây mô tả các thành phần quan trọng của Node.js:



### Node.js được sử dụng ở đâu

Dưới đây là các lĩnh vực mà Node.js được sử dụng như là một sự lựa chọn hoàn hảo:

* Các ứng dụng về I/O.
* Các ứng dụng về luông dữ liệu.
* Các ứng dụng về dữ liệu hướng đến thời gian thực.
* Các ứng dụng dựa vào JSON APIs.
* Các ứng dụng Single Page Application.

### Node.js không nên sử dụng ở đâu

Node.js không nên sử dụng trong các ứng dụng đòi hỏi về CPU.

# Cài đặt

## Dependencies

### NodeJS

NodeJs có thể coi như là nền móng của tất cả các Javascript application trên server. Việc cài đặt NodeJS có thể thực hiện đơn giản trên các hệ điều hành khác nhau:

* Ubuntu với apt-get



* MacOS với homebrew



* Các hệ điều hành khác có thể tìm các gói cài đặt tại trang chủ của NodeJS: <https://nodejs.org>

### MongoDB

MongoDB là một NoSQL database điển hình, các ứng dụng hiện đại đang dịch chuyển dần từ việc sử dụng MySQL sang NoSQL database và MeanJS không nằm ngoài xu thế đó. Việc cài đặt MongoDB cũng giống như node, có thể thực hiện đơn giản bằng “apt get” hoặc “brew”.

### Bower và Grunt





## Cài đặt MeanJS

Hướng dẫn cụ thể cho việc cài đặt MeanJS có thể tìm thấy trên trang chủ của MeanJS “ <https://meanjs.org> ” hoặc trên Github “ <https://github.com/meanjs/mean>”

Về cơ bản chúng ta có thể khởi tạo một MeanJS application bằng việc “clone” trực tiếp trên Github. Tuy nhiên, chúng ta có thể tạo MeanJS project dùng “Yeoman” như sau:

Giống như các framework khác đều cung cấp các công cụ để có thể tạo scaffold, chúng ta có thể khởi tạo các files dành cho Mean dùng “Yeoman”.

Cài đặt Yeoman rất đơn giản, chỉ với:



Sau đó, chúng ta cài đặt công cụ Generate cho MeanJS



Để khởi tạo MeanJS project sử dụng “Yeoman” chúng ta chỉ cần:



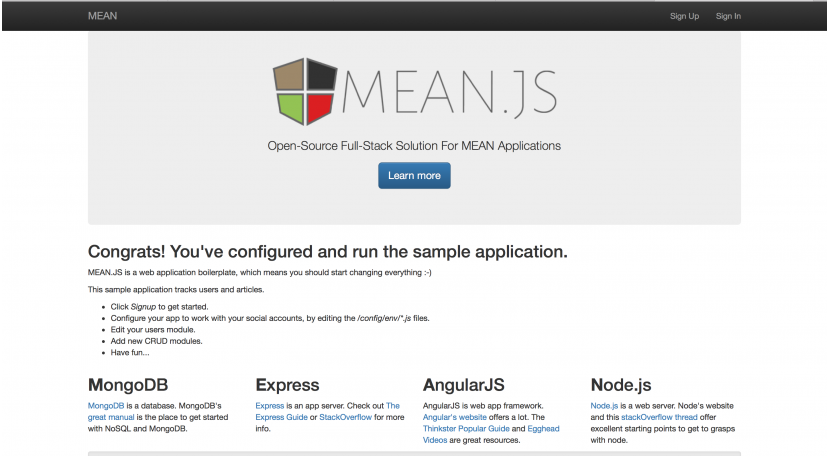
Lúc này trên “Yeoman” sẽ đưa cho chúng ta các lựa chọn cho project, tùy vào yêu cầu của ứng dụng mà lựa chọn các gói cho phù hợp.



Sau khi cài đặt các gói thành công, chúng ta hoàn toàn có thể kiếm tra xem MeanJS đã cài thành công chưa khi khởi tạo server bằng grunt



Truy cập vào local server (localhost:3000), nếu như MeanJs project được khởi tạo thành công, màn hình sẽ trả về như sau:



# Tìm hiểu MEANJS qua việc làm ứng dụng viết blog

Hôm nay, chúng ta cùng tìm hiểu kỹ hơn về bộ MEAN thông qua việc viết 1 ứng dụng viết blog sử dụng bộ 4 công nghệ này (MongoDB, ExpressJS, AngularJS, NodeJS).

Trong bài này tôi sử dụng:

* NodeJS
* ExpressJS
* MongoDB
* Mongoose
* Jade

Mục đích của bài này tối muốn chia sẻ với các bạn:

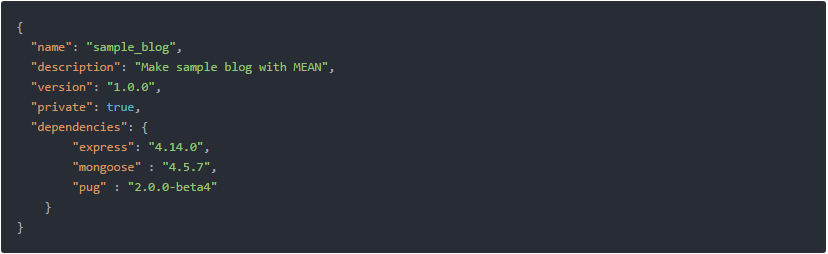
* Làm sao để xây dựng một trang web động sử dụng nodejs.
* Kết hợp nodejs và mongodb như thế nào.
* Export module trong nodejs.

## Cấu trúc folder của ứng dụng



## File package.json

Tại đây ta nhập các thông tin về ứng dụng chúng ta làm

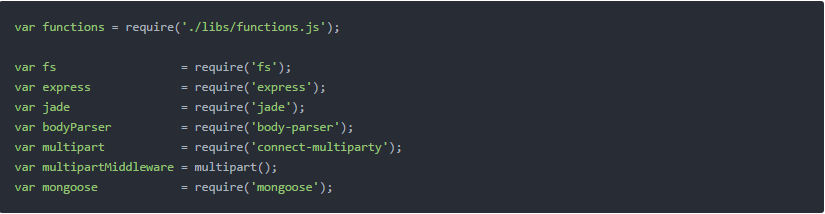


Sau đó ta cài các bộ này bằng lệnh: “npm install”

## Load các Module cần thiết

Việc đầu tiên và rất quan trọng đó là load các module bạn muốn sử dụng.

File để chạy ứng dụng của chúng ta là “server.js”



## Cấu hình cho ứng dụng



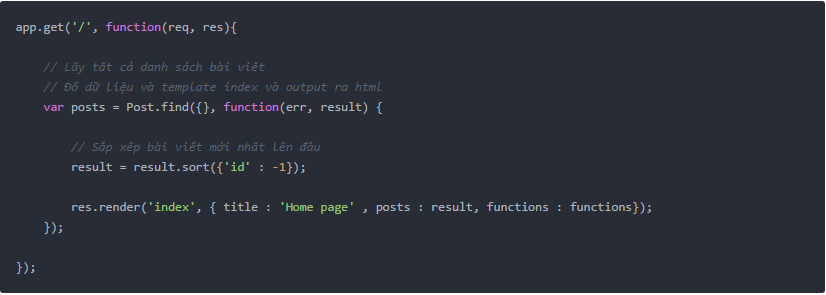
## Kết nối đến MongoDB



## Các trang chính

Trong bài này, tồi làm 3 trang danh sách các blogs, chi tiết và trang tạo blog. Trang danh sách sẽ liệt kê tất cả các blogs, sau đó click vào một trong các blog đó sẽ ra trang chi tiết nội dung cụ thể của blog đó. Và trang tạo bài viết để có thể viết một bài mới sau đó bổ sung vào trang danh sách.

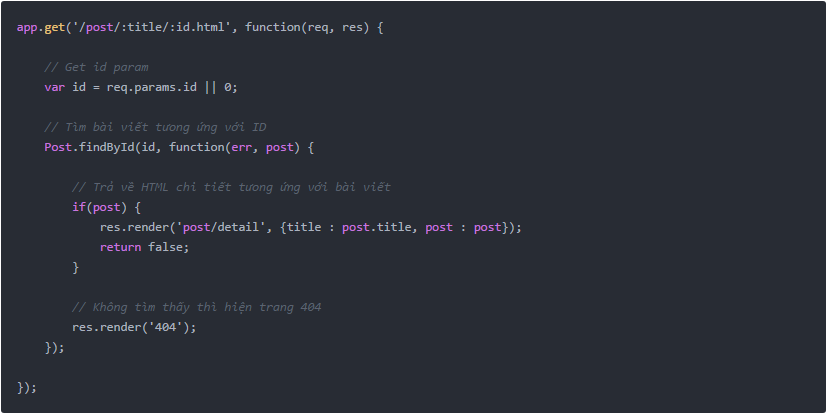
### Trang danh sách Blogs



Và file index.jade là



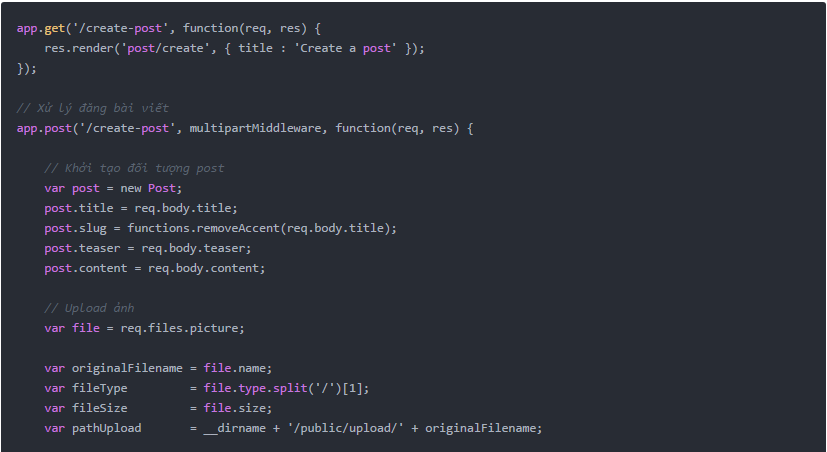
### Trang chi tiết



Và file index.jade là



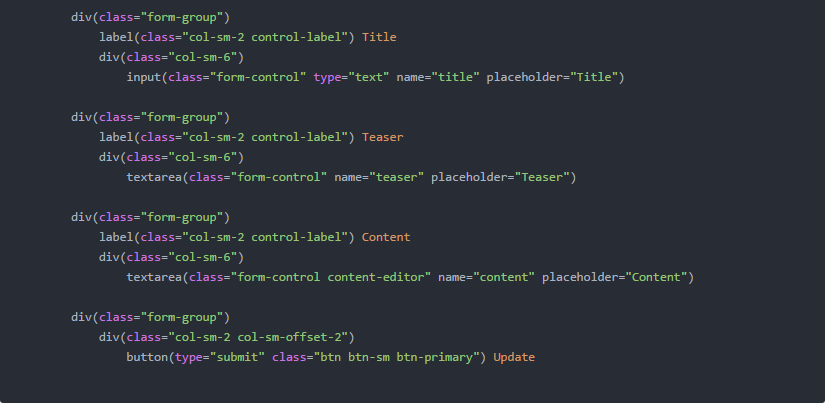
### Trang tạo blog





Và giao diện tạo blog ở file index.jade là





Chúng ta chạy bằng lệnh “node server.js”

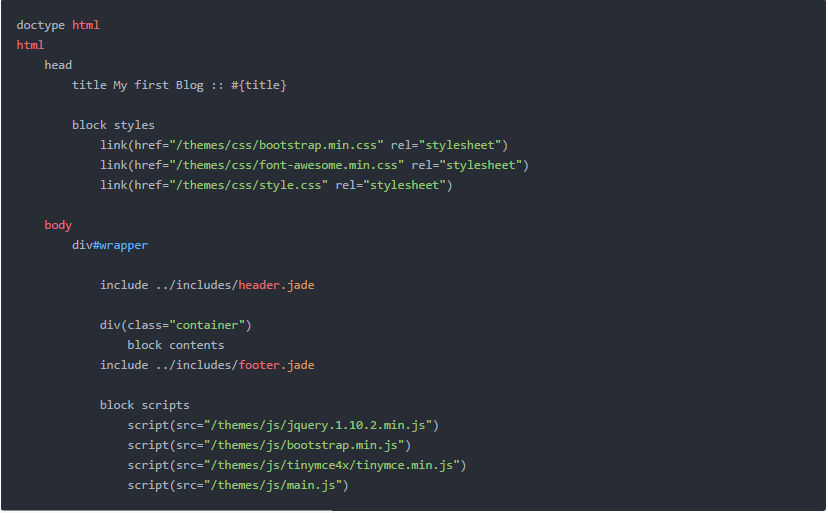
## Tại sao chọn Jade template để tạo giao diện cho ứng dụng

Chúng ta sẽ sử dụng jade làm template để xây dựng lên giao diện blog. Jade có cơ chế extends template rất linh hoạt, nếu ai đã từng sử dụng Laravel (PHP framework) thì chắc chắn sẽ thích điều này.

Khi sử dụng jade chúng ta không cần quan tâm tới thẻ đóng html, cú pháp cực kì gọn, dễ đọc, dễ hiểu.

Khi template con extends một temlate cha thì mặc định nó sẽ có hết nội dung từ template cha.

Để cho dễ hiểu hãy xem một file jade mà tôi tạo ra như sau:

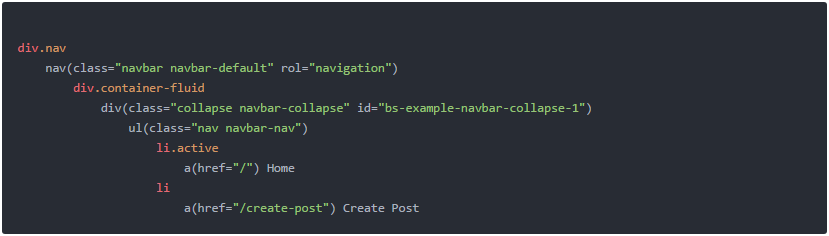


Tôi có định nghĩa ra 3 khối đó là ‘block styles, block scripts, block contents’.

* Block styles: nơi để template con có thể thêm những style riêng của mình vào ngoài những style mà template cha quy định.
* Block scripts: nơi để template con có thể thêm những đoạn script riêng của mình vào ngoài những script mà template cha quy định.
* Block contents: nơi chứa nội dung mà template con định nghĩa ra.

Dùng include để chèn nội dung của file template khác vào 1 file. (giống @include trong Laravel).

Trong bài này, chúng ta có include 2 file đó là:



Và



# Kết luận

Qua bài viết chung ta tìm hiểu cơ bản về cách làm việc của MEAN, chủ yếu ở đây là tìm hiểu về làm việc với MongoDB thông qua “mongoose” và tạo template bằng “Jade template”. Bài viết trình bày ở mức cơ bản nên có thể có nhiều thiết sót cũng như chưa được tối ưu.

MeanJS ra đời tuy chưa lâu nhưng thực ra đang tạo ra một cơn sốt mới dành cho những nhà phát triển ứng dụng. Với MeanJS, ngoài việc tận dụng được hiệu năng cực kỳ tốt của các ứng dụng Javascript, MeanJS còn cho phép tùy biến khá tốt. Với MeanJS, ta có thể xây dựng một trang web hoàn thiện sử dụng Javascript hoặc sử dụng như một End user service point của một Data server thuần túy.

# Tài liệu tham khảo

1. <http://vietjack.com/mongodb/tong_quan_ve_mongodb.jsp>

2. <http://vietjack.com/nodejs/express_framework_trong_nodejs.jsp>

3. <http://vietjack.com/angularjs/angularjs_la_gi.jsp>

4. <http://vietjack.com/nodejs/nodejs_la_gi.jsp>

5. <https://viblo.asia/ducthien1490/posts/YrEBRAjqM8Zj#back-end-11>

6. <https://viblo.asia/huynguyen/posts/PdbknxzNGyA>