

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN: LẬP TRÌNH WEB VÀ ỨNG DỤNG

TÌM HIỂU NODE JS

Người hướng dẫn: **G.V Đặng Minh Thắng**

Người thực hiện: **Trần Quốc Lĩnh - 51703124**

Lớp: 17050301

Khoá: 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN: LẬP TRÌNH WEB VÀ ỨNG DỤNG

TÌM HIỂU NODE JS

Người hướng dẫn: **G.V Đặng Minh Thắng**

Người thực hiện: **Trần Quốc Lĩnh - 51703124**

Lớp: 17050301

Khoá: 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

LỜI CẢM ƠN

Cảm ơn thầy Đặng Minh Thắng đã dạy môn lập trình web và ứng dụng hết sức nhiệt tình và tâm huyết, cảm ơn thầy đã định hướng và tạo cơ sở để em có thể tự tìm hiểu và học thêm về cách lập trình và tổ chức một trang web.

Còn đây là bài tập lớn, một nội dung trong trường trình giảng dạy mà thầy đã giao cho em. Trong quá trình làm bài tập lớn này, em vẫn còn nhiều thiếu sót. Em rất mong nhận được sự đánh giá và chỉ bảo từ thầy!

BÀI TẬP LỚN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Em xin cam đoan đây là sản phẩm bài tập lớn của riêng em. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu, hình ảnh được chính em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài tiểu luận còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào em xin hoàn toàn chịu trách nhiệm về nội dung bài tập lớn của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do em gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 31 tháng 03 năm 2019

Tác giả

(Đã ký)

Trần Quốc Lĩnh

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày 31 tháng 03 năm 2019
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày 31 tháng 03 năm 2019
(kí và ghi họ tên)

TÓM TẮT

Internet đã xuất hiện hơn một thập kỷ qua, và sự phổ biến của nó đã được lan rộng trên toàn cầu. Nên là hầu như trong mỗi chúng ta ai ai cũng biết một trang web trông thế nào, có nhiều loại ra sao. Nhưng không nhiều người biết đến cách một trang web được tạo ra và vận hành như thế nào. Và trong bài tập lớn này, em xin trình bày một nội dung, một phương thức dùng để quản lí và vận hành server của web. Node JS

Gồm có các nội dung chính:

- Tổng quan về Node JS
- Thiết kế một web chat đơn giản bằng Node JS

MỤC LỤC

LỜI CẢM ƠN	3
CAM KẾT	4
ĐÁNH GIÁ CỦA GIÁO VIÊN	5
TÓM TẮT	6
MỤC LỤC	7
DANH MỤC CHÚ THÍCH CÁC THUẬT NGỮ VÀ HÌNH ẢNH	8
TỔNG QUAN VỀ NODE JS	9
1. Node JS là gì?	9
2. Đặc điểm của Node JS?	10
3. Cơ chế hoạt động	10
4. Ứng dụng của Node JS	11
5. Ưu điểm và nhược điểm	11
THIẾT KẾ MỘT WEB CHAT ĐƠN GIẢN BẰNG NODE JS	12
1. Cách cài đặt và chạy demo	12
2. Mô tả code	18
TÀI LIỆU THAM KHẢO	21

DANH MỤC CHÚ THÍCH CÁC THUẬT NGỮ VÀ HÌNH ẢNH

Thuật ngữ

Heap : Là vùng nhớ được dùng để chứa kết quả tạm phục vụ cho việc thực thi các hàm trong stack. Heap càng lớn thì khả năng tính toán càng cao.

Stack : Là một kiểu hàng đợi first in last out - vào trước ra sau.

Js runtime engine : Là khu thực thi câu lệnh js gồm Heap và stack.

Web APIs : Là nơi chức các tác vụ được cung cấp bởi trình duyệt.

Callback queue : Là hàng đợi công việc kiểu first in first out - vào trước ra trước.

Event Loop : Đọc Stack và Event Queue. Nếu nhận thấy Stack rỗng nó sẽ nhấc Event đầu tiên trong Event Queue và handler (callback hoặc listener) gắn với Event đó và đẩy vào Stack. Đặc điểm của việc thực thi hàm trong JS là sẽ chỉ dừng lại khi hàm return hoặc throw exception. Có nghĩa là trong khi hàm đang chạy thì sẽ không có một hàm khác được chạy, dữ liệu tạm của hàm cũng sẽ không bị thay đổi bởi một hàm khác hay cũng không bị dừng lại cho đến khi hoàn thành.

Hình ảnh

Hình 1: Các định nghĩa của Node JS

Hình 2: Cơ chế hoạt động

Hình 3: Download Node JS

Hình 4: Cách cài đặt - Chọn next

Hình 5: Cách cài đặt - Tick vào ô I accept và chọn next

Hình 6: Cách cài đặt - Chọn thư mục để cài rồi chọn next

Hình 7: Cách cài đặt - Chọn next

Hình 8: Cách cài đặt - Chọn next

Hình 9: Cách cài đặt - Đợi cho đến khi việc cài đặt hoàn tất

Hình 10: Cách cài đặt - Chọn finish

Hình 11: Cài đặt module cho Node JS để chạy realtime

Hình 12: Lệnh thực thi trên cmd

Hình 13: Trên cửa sổ trình duyệt

Hình 14: Demo chat real time

Hình 15: Khởi tạo biến và import module cần thiết

Hình 16: Load một trang html từ file

Hình 17: Tạo một server lắng nghe trên cổng 8000

Hình 18: Tạo biến kết nối với server trên cổng 8000

Hình 19: Client giao tiếp với server

Hình 20: Server handle các sự kiện client gửi tới

TỔNG QUAN VỀ NODE JS

1. Node JS là gì?

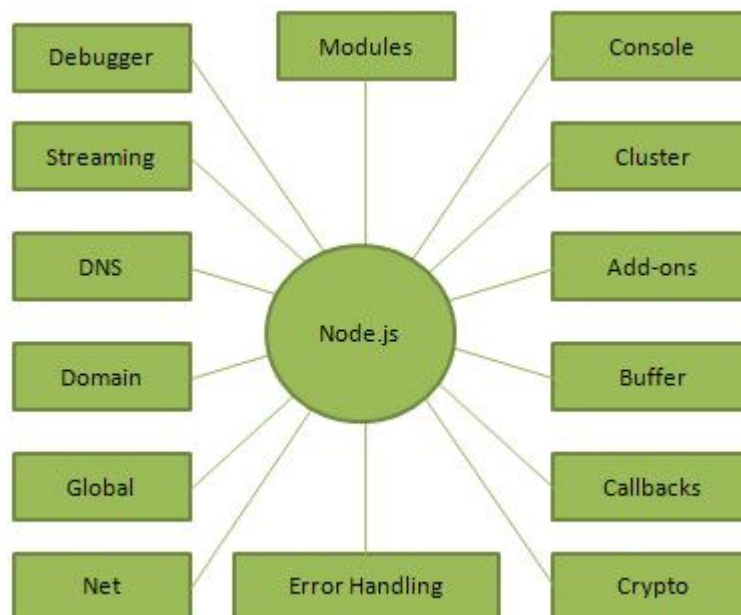
Node js là một mã nguồn mở, một môi trường, một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh, theo thời gian thực chạy trên trình duyệt Chrome.

Phần Core bên dưới của Node js được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Node js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS.

Mọi hàm trong Node js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing)

Node.js được tạo bởi Ryan Dahl từ năm 2009, và phát triển dưới sự bảo trợ của Joyent.

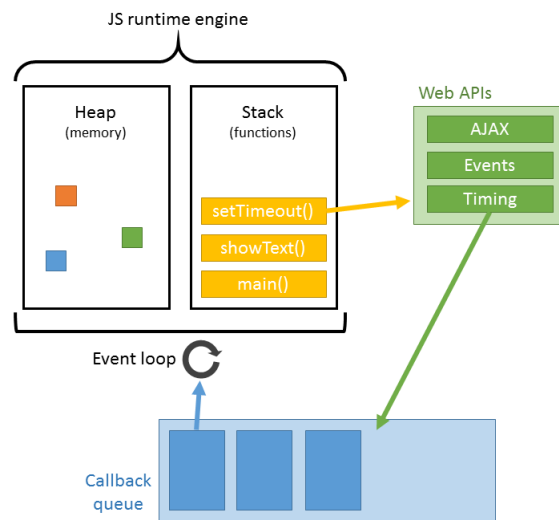


Hình 1: Các định nghĩa của Node JS

2. Đặc điểm của Node JS

- Node JS là không phải là một ngôn ngữ mà nó chỉ là môi trường để chạy javascript phía server.
- Node JS nó có tất cả các tính chất của javascript.
- Node JS xử lý bất đồng bộ.
- Node JS chạy đơn luồng. Tức là Node JS chạy trên 1 CPU core duy nhất.
- Là ngôn ngữ hướng sự kiện.

3. Cơ chế hoạt động



Hình 2: Cơ chế hoạt động

B1: Đầu tiên các câu lệnh được chạy lần lượt từ trên xuống dưới và đưa vào hàng đợi queue.

B2: Vì lúc này stack vẫn trống nên event loop sẽ lấy một tác vụ ở trong queue bỏ vào stack.

B3: Sau khi xử lý xong thì tác vụ đó sẽ được lấy ra khỏi stack, và cho tác vụ tiếp theo vào. Nếu ta thấy có một hàm nào đó trong web APIs dùng để tính toán thời gian, nó sẽ được đưa vào Web APIs để đợi.

B4: Thực hiện các tác vụ còn lại tương tự như trên, cho đến khi hàng đợi rỗng, thì tác vụ đang đợi ở web APIs sẽ được đưa vào stack để thực hiện.

4. Ứng dụng của Node JS

- Websocket server: Chat Online, Game online...
- Fast File Upload Client: Các chương trình upload file tốc độ cao.
- Ad server: Máy chủ quảng cáo.
- Cloud services: Các dịch vụ đám mây.
- RESTful API: Dùng cho các ứng dụng thông qua API.
- Any Real-time Data Application: Các ứng dụng yêu cầu tốc độ thời gian thực.

5. Ưu điểm và nhược điểm

* Ưu điểm:

- Xử lý được nhiều yêu cầu chỉ bằng đơn luồng (single thread). Giúp hệ thống đỡ tốn tài nguyên và tối ưu tốc độ thực thi, giảm độ trễ.
- Có cơ chế event-driven, và mô hình kết hợp với Javascript đáp ứng tốt cho các dịch vụ web.
- Có khả năng xử lý được rất nhiều request mà không cần tải lại trang và khả năng phản hồi nhanh.
- Node JS xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
- Rất hiệu quả khi xây dựng những ứng dụng thời gian thực.

* Nhược điểm:

- Khi cần sử dụng những ứng dụng tốn tài nguyên CPU như encoding cho video, convert file,... hay tương tự làm cho chương trình nặng nề tốn tài nguyên. Hiệu suất hoạt động kém.
- Node JS còn sơ khai và nhiều mặt vẫn chưa đáp ứng tốt.

THIẾT KẾ WEB CHAT ĐƠN GIẢN BẰNG NODE JS

1. Cách cài đặt và chạy demo.

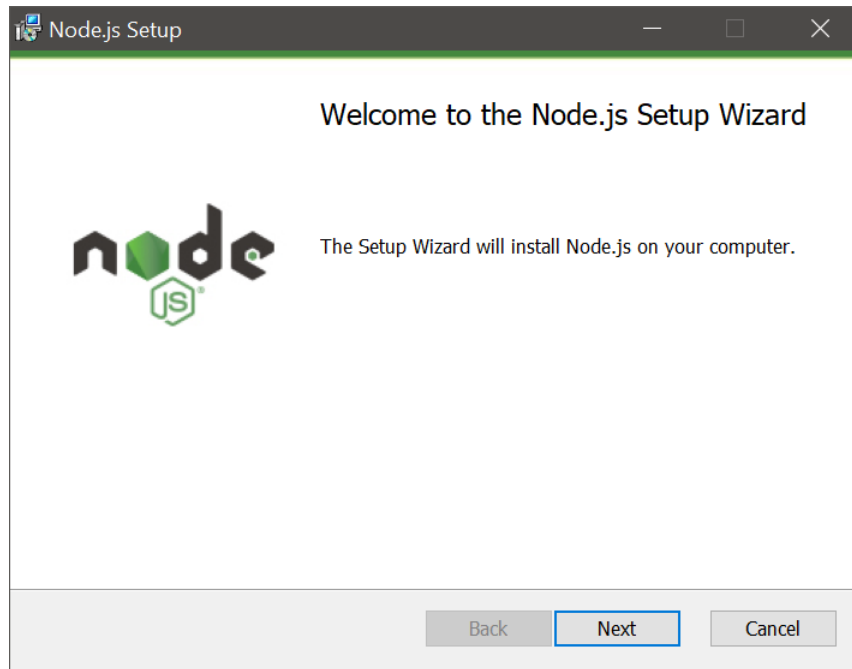
Để sử dụng được thì trước hết ta phải cài đặt Node JS vào máy. Truy cập trang web dưới đây theo đường link <https://nodejs.org/en/download/> để tiến hành download và cài đặt. Ở đây chọn mục Windows Installer(.msi) 64-bit.

The screenshot shows the Node.js download page. At the top, there's a navigation bar with the URL <https://nodejs.org/en/download/>. Below the navigation bar, the page is titled "Downloads" and mentions the "Latest LTS Version: 10.15.3 (includes npm 6.4.1)". A message encourages downloading the Node.js source code or a pre-built installer. The page is divided into two main sections: "LTS Recommended For Most Users" and "Current Latest Features". Under the "LTS" section, there are three download options: "Windows Installer" (node-v10.15.3-x64.msi), "macOS Installer" (node-v10.15.3.pkg), and "Source Code" (node-v10.15.3.tar.gz). Below these, there is a list of available binaries: "Windows Installer (.msi)", "Windows Binary (.zip)", "macOS Installer (.pkg)", "macOS Binary (.tar.gz)", "Linux Binaries (x64)", and "Linux Binaries (ARM)". To the right of this list is a table showing the available binaries for different architectures.

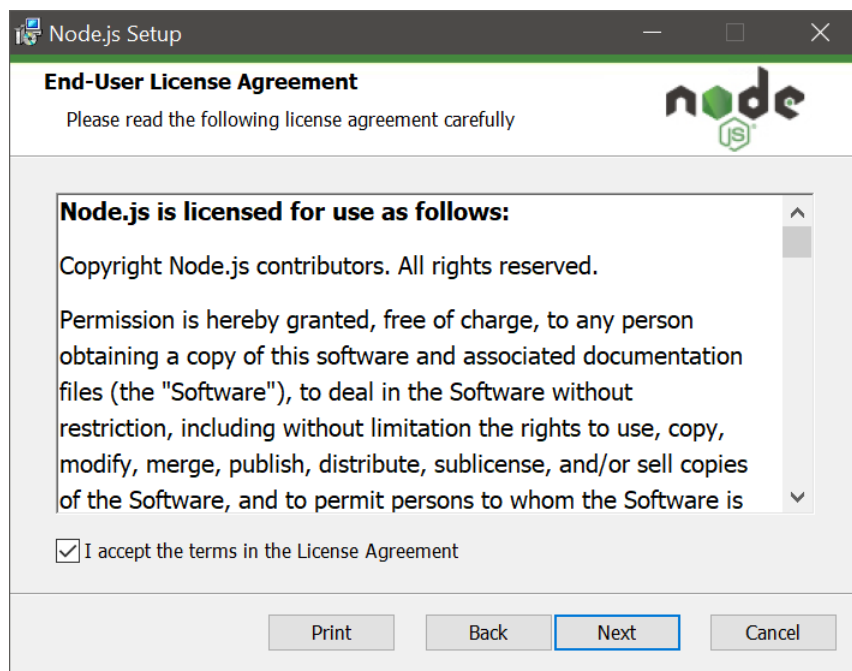
32-bit		64-bit	
32-bit		64-bit	
64-bit			
64-bit			
64-bit			
ARMv6		ARMv7	
		ARMv8	

Hình 3: Download Node JS

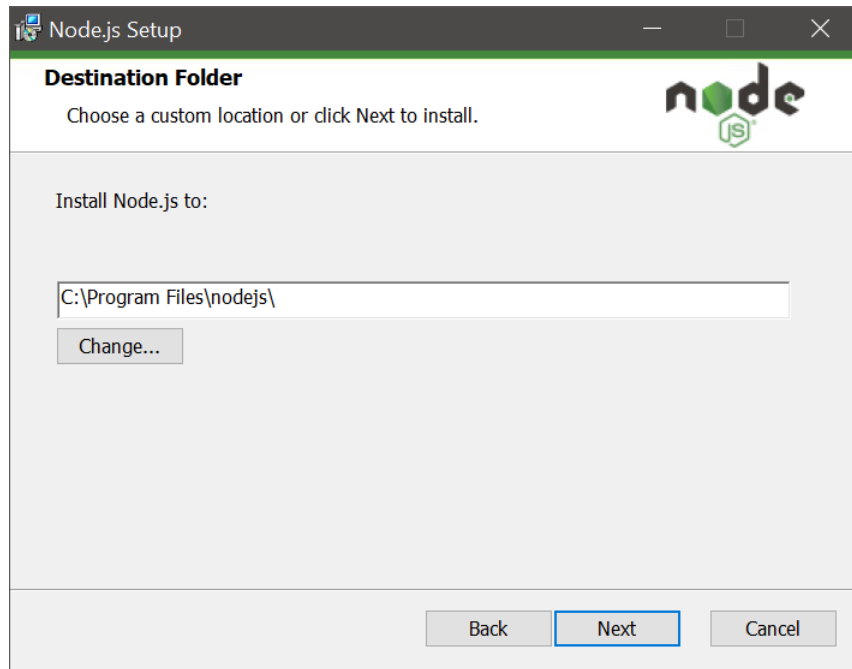
Sau khi tải xong, chúng ta mở file đó để bắt đầu cài lên máy. Lần lượt theo cái bước sau.



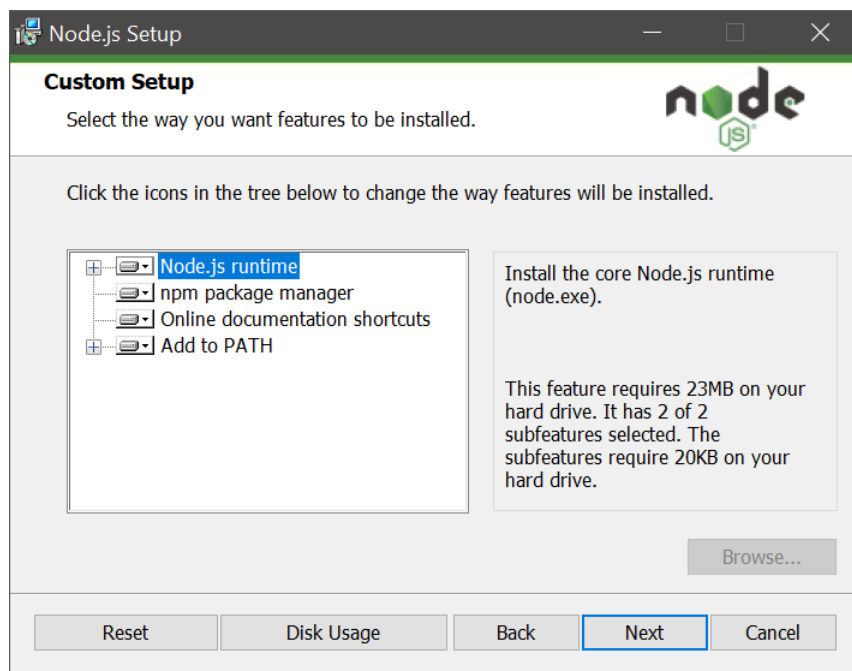
Hình 4: Cách cài đặt - Chọn next



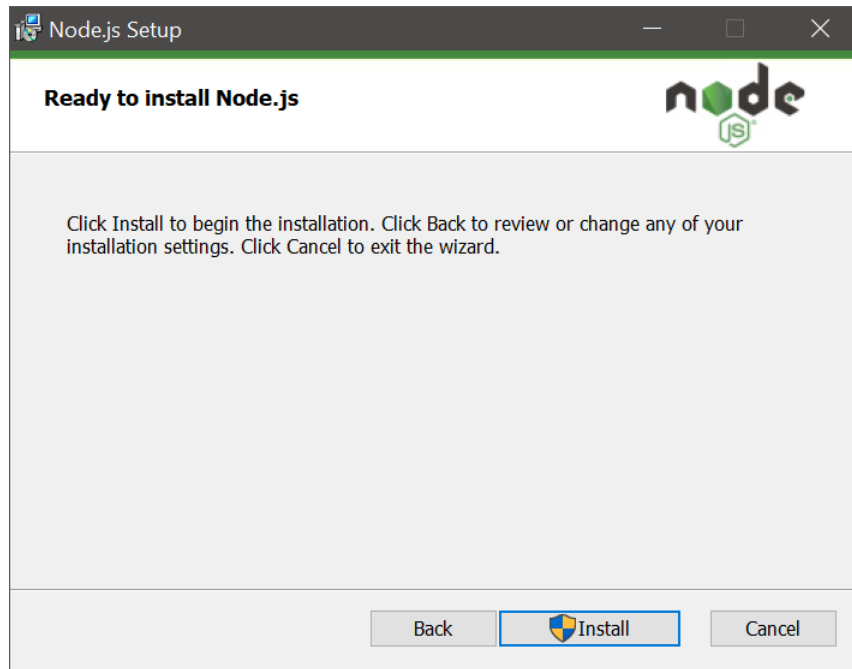
Hình 5: Cách cài đặt - Tick vào ô I accept và chọn next



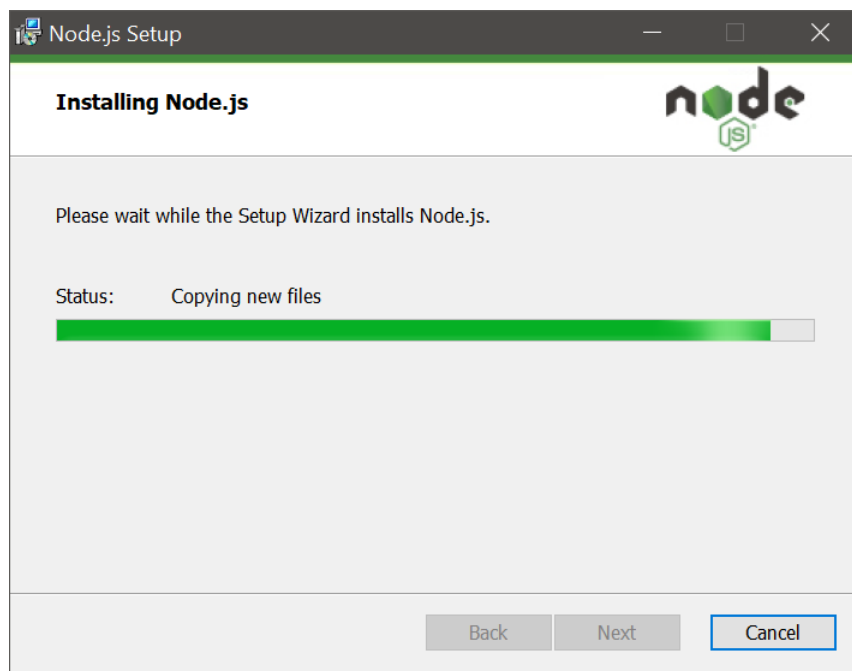
Hình 6: Cách cài đặt - Chọn thư mục để cài rồi chọn next



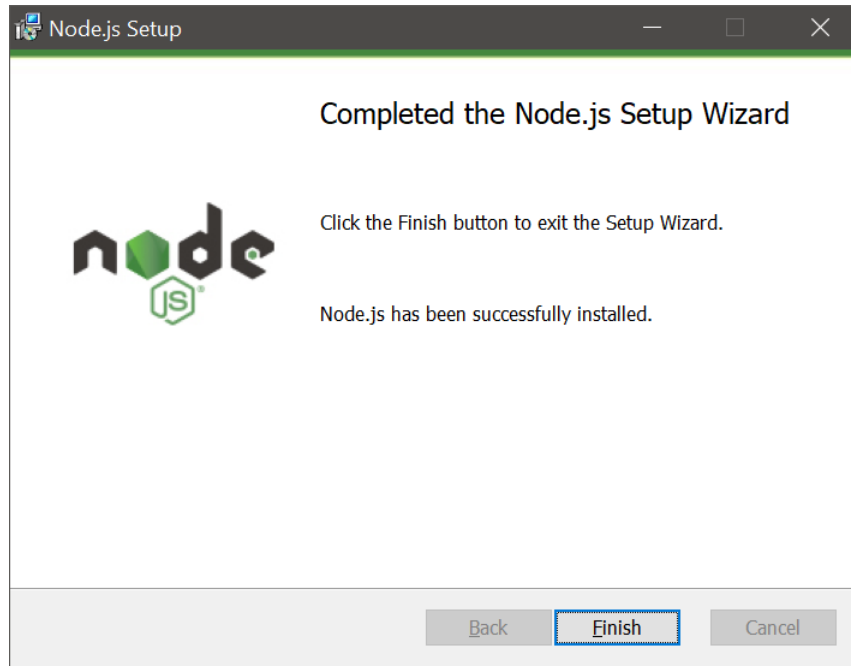
Hình 7: Cách cài đặt - Chọn next



Hình 8: Cách cài đặt - Chọn next



Hình 9: Cách cài đặt - Đợi cho đến khi việc cài đặt hoàn tất



Hình 10: Cách cài đặt - Chọn finish

Vậy là việc cài đặt node js đã hoàn tất, kể đến chúng ta cần cài đặt thêm một số module cần thiết để chạy được code demo. Mở cửa sổ cmd lên và dẫn tới thư mục chứa code demo và chạy câu lệnh sau

"npm install socket.io" --save (Đây là đoạn lệnh cài đặt module socket.io hỗ trợ cho việc chạy realtime)

```
C:\Users\memor\Desktop\ChatRealTime>npm install socket.io --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN chatrealtime@1.0.0 No description
npm WARN chatrealtime@1.0.0 No repository field.

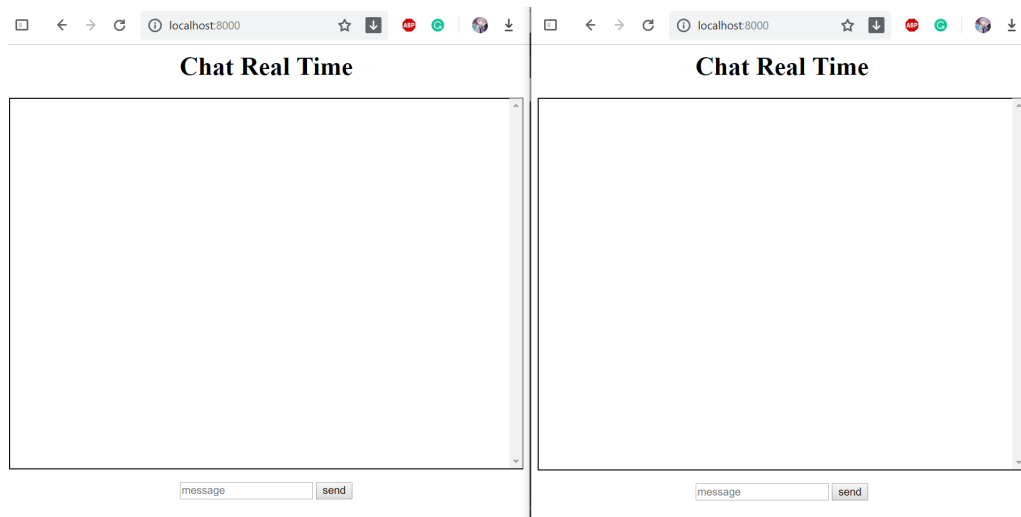
+ socket.io@2.2.0
added 45 packages from 33 contributors and audited 77 packages in 9.971s
found 0 vulnerabilities
```

Hình 11: Cài đặt module cho Node JS để chạy realtime

Bây giờ ta tiến hành chạy code demo, dùng lệnh "node server.js" trên cmd để chạy và mở trang 2 tab của trình duyệt và truy cập đường dẫn <http://localhost:8000/> để quan sát việc chạy realtime của node JS thông qua một ứng dụng chat đơn giản.

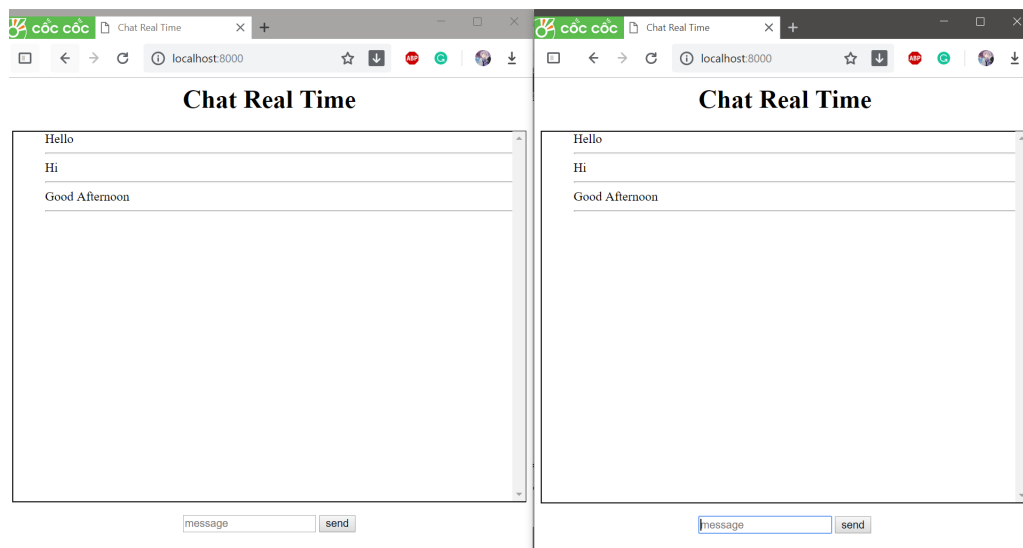
```
C:\Users\memor\Desktop\ChatRealTime>node server.js
Client connecting...
Hello server
Client connecting...
Hello server
```

Hình 12: Lệnh thực thi trên cmd



Hình 13: Trên cửa sổ trình duyệt

Người dùng tự nhập một số đoạn văn bản ngắn vào ô message rồi click vào nút send trên cả hai tab để quan sát tính realtime mà node js mang lại, dưới đây là ví dụ mẫu.



Hình 14: Demo chat real time

2. Mô tả code.

Trong phần code demo gồm có:

- 1 file server.js
- 1 file package.json lưu trữ các thông tin.
- 1 thư mục public chứa: 1 file index.html, 1 file client.js, 1 file style.css.

*Đối với server

a) Khởi tạo

```
var express = require("express");
var app = express();

var server = require('http').createServer(app);
var io = require('socket.io')(server);
```

Hình 15: Khởi tạo biến và import module cần thiết

b) Load trang web html

```
app.get('/', function (req, res, ext) {  
  res.sendFile(__dirname + '/public/index.html');  
});
```

Hình 16: Load một trang html từ file

c) Tạo một server

```
server.listen(8000);
```

Hình 17: Tạo một server lắng nghe trên cổng 8000

*Đối với client

a) Khởi tạo

```
var socket = io.connect('http://localhost:8000');
```

Hình 18: Tạo biến kết nối với server trên cổng 8000

*Giao tiếp giữa server và client

a) Phía client

- Client handle sự kiện connect và gửi một sự kiện join lên server.
- Client handle sự kiện thread và thêm một thông tin vào trang html.
- Client handle sự kiện submit của form và gửi dữ liệu về cho server qua sự kiện message.

```

socket.on('connect', function(data){
    socket.emit('join','Hello server');
});

socket.on('thread', function(data){
    $('#thread').append('<li>' + data + '</li>' + '<hr>');
});

$('form').submit(function(){
    var message = $('#message').val();
    socket.emit('message', message);
    this.reset();

    return false;
})

```

Hình 19: Client giao tiếp với server

b) Phía server

- Server handle sự kiện connect.
- Server handle sự kiện join.
- Server handle sự kiện message và gửi broadcast cho tất cả biết là có một sự kiện thread vừa được tạo ra kèm theo data.

```

io.on('connection', function(client){
    console.log('Client connecting...');
    client.on('join', function(data){
        console.log(data);
    });
    client.on('message', function(data){
        client.emit('thread', data);
        client.broadcast.emit('thread', data);
    });
});

```

Hình 20: Server handle các sự kiện client gửi tới

TÀI LIỆU THAM KHẢO

Tiếng Việt

<https://vi.wikipedia.org/wiki/Node.js>

<https://viblo.asia/p/tim-hieu-ve-node-js-co-ban-ojaqG0dGEKwZ>

<https://techtalk.vn/mot-cai-nhin-tong-quan-nhat-ve-nodejs.html>

<https://techblog.vn/tim-hieu-ve-node-js-co-ban>

Tiếng Anh

<https://nodejs.org/en/docs/>