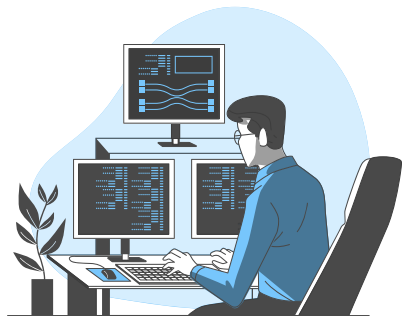


# Khóa học Backend

## Bài 14: NPM, NodeJS, Express (Tiết 1)



# Nội dung

01

Khái niệm NodeJS

...

02

NPM

...

03

ExpressJS

...

04

PUG

...





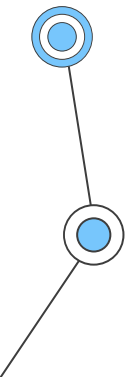
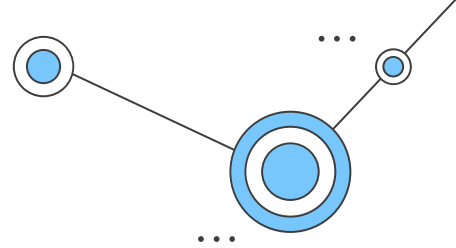
# 01

## Khái niệm NodeJS



# 1. Khái niệm NodeJS

- **NodeJS** là mã nguồn mở, dùng để **chạy Javascript đa nền tảng**.
- **Trước đây**, Javascript chỉ dùng để chạy ở môi trường là trình duyệt bên phía Front-end.
- **Hiện nay**, NodeJS sinh ra đóng vai trò làm môi trường để chạy được Javascript ở môi trường khác trình duyệt.
- Ví dụ như code Back-end sử dụng Javascript.
- Download và cài đặt: <https://nodejs.org/>
- Kiểm tra version NodeJS: **node -v**



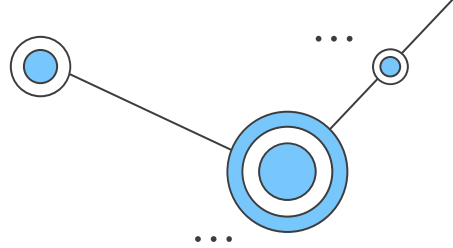


02

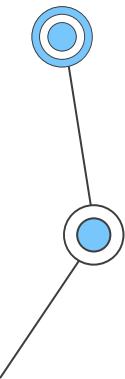
NPM



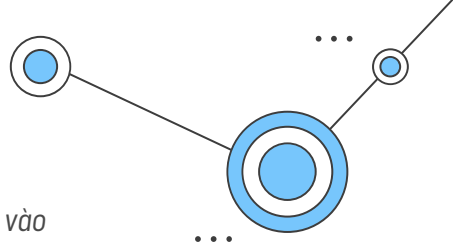
## 2.1. Khái niệm



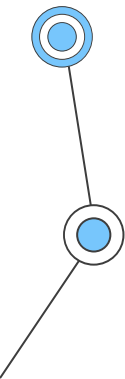
- **NPM** viết tắt của **N**ode **P**ackage **M**anager.
- Là một **công cụ** tạo và **quản lý các thư viện** lập trình **Javascript** cho Node.js.
- *Không sử dụng NPM, bạn sẽ cần tải toàn bộ các thư viện một cách thủ công. Sau đó, bạn cần thực hiện nhúng thư viện vào chính dự án của mình. Điều này làm mất nhiều thời gian hơn để hoàn thành.*
- *Khi sử dụng NPM, bạn chỉ cần 1 dòng lệnh hoàn tất công việc lưu thư viện.*
- Link website: <https://www.npmjs.com/>



## 2.2. Cài đặt và sử dụng npm

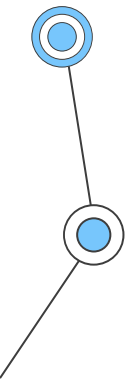
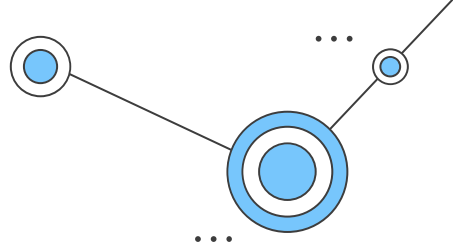


- **Bước 1:** Gõ lệnh **npm -v** trên **cmd** để kiểm tra phiên bản.
  - (Npm được tích hợp sẵn trong nodejs, nên chỉ cần cài đặt nodejs là được, nếu chưa cài đặt thì vào trang <https://nodejs.org/en/> để cài đặt).
- **Bước 2:** Gõ lệnh **npm init**, sau đó cứ bấm enter đến hết.
  - Câu lệnh này sẽ **tạo ra 1 file** có tên là **package.json** – file này để lưu trữ thông tin (tên package, phiên bản, các dependencies) mà project của bạn sử dụng.
- **Bước 3:** Gõ lệnh **npm install [tên package]** hoặc **npm i [tên package]**. Để cài đặt package vào dự án.



## 2.3. Một số thuộc tính trong package.json

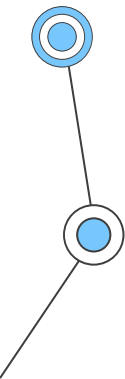
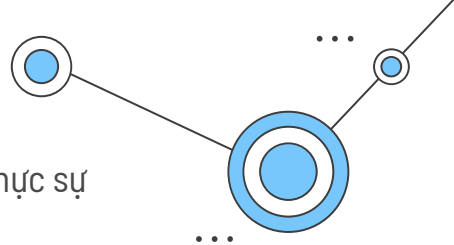
- **name:** tên gói thư viện.
- **version:** phiên bản gói.
- **description:** phần mô tả về gói thư viện.
- **homepage:** trang chủ của gói.
- **author:** tác giả.
- **contributors:** tên người đóng góp cho package.
- **dependencies:** danh sách các gói phụ thuộc, tự động được cài theo.
- **repository:** loại repository và url của package, thông thường là link git.
- **main:** index.js (file chính của dự án).
- **keywords:** các từ khóa.





## 2.4. Một số câu lệnh npm khác

- **npm update [tên\_package]:** Để update một package. Hạn chế dùng, chỉ dùng khi thực sự hiểu về package bạn đang muốn update.
- **npm uninstall [tên\_package]:** Gỡ cài đặt package.



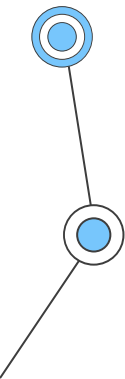
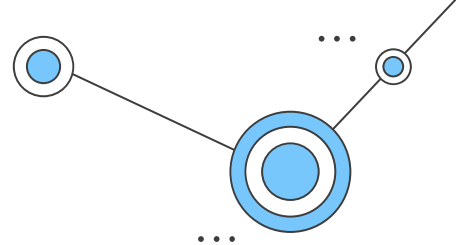
03

ExpressJS

## 3.1. Khái niệm ExpressJS

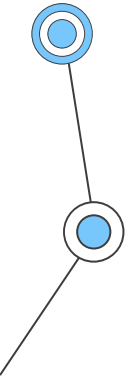
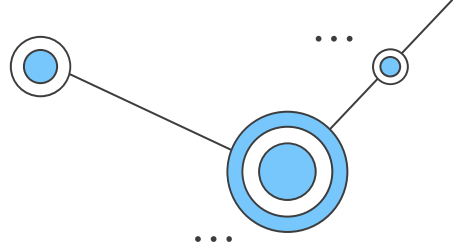
- **ExpressJS** là một framework để **xây dựng các ứng dụng** web trong **NodeJS**.
- **ExpressJS** giúp quá trình xây dựng server trở nên **đơn giản** với **nhiều tính năng** được xây dựng sẵn.
- Cài đặt: <https://www.npmjs.com/package/express>

Express 

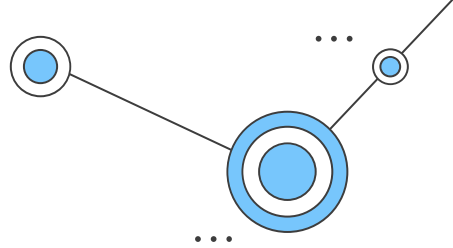


## 3.2. Ví dụ Hello World!

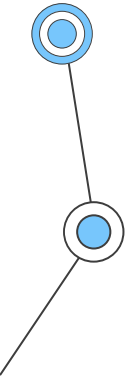
- Bước 1: Tạo một folder **vi-du-01** và chạy **npm init**
- Bước 2: Cài đặt express: **npm i express**
- Bước 3: Tạo file **index.js** và thêm code
- Bước 4: Bật Terminal lên và khởi chạy bằng câu lệnh **node index.js**



## 3.3. Cài Nodemon

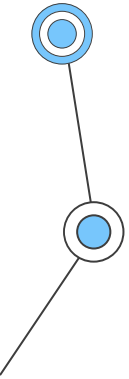
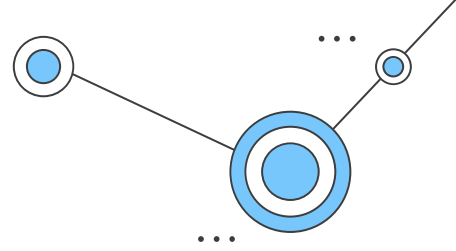


- **Nodemon** là một công cụ phục vụ cho NodeJS.
- Giúp **tự động khởi động lại ứng dụng** khi phát hiện các thay đổi của tệp.
- Link NPM: <https://www.npmjs.com/package/nodemon>
- Hướng dẫn cài đặt:
  - Bước 1: Chạy câu lệnh **npm i --save-dev nodemon**
  - Bước 2: Thêm dòng **"start": "nodemon --inspect index.js"** vào mục **script** trong file **package.json**
  - Bước 3: Chạy **npm start**

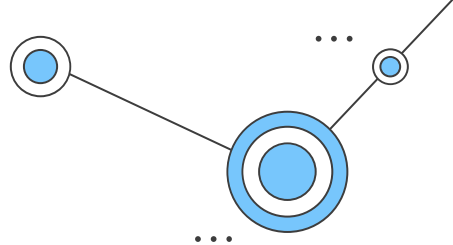


## 3.4. Sử dụng Routing cơ bản

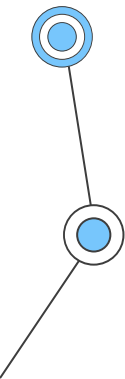
- *Hướng dẫn trong lúc học.*



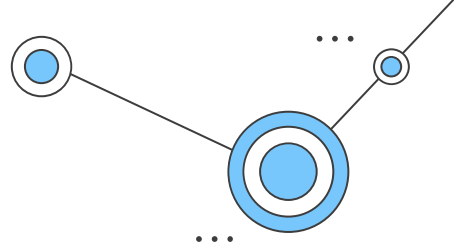
## 3.5. SPA và MPA



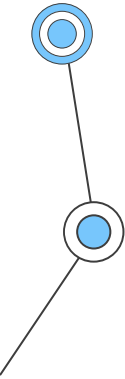
- **Multiple Page Application** là gì?
  - **MPA** là khi người dùng **truy cập các trang con** trên web, thì server sẽ xử lý và **trả về toàn bộ trang web** và web đó sẽ được **load lại mới hoàn toàn**.
  - Ví dụ: shopee.vn, tiki.vn, sendo.vn,...
- **Single Page Application** là gì?
  - **SPA** là khi người dùng **truy cập các trang con** trên web, website sẽ **chỉ load lại những phần giao diện thay đổi**, những giao diện không thay đổi sẽ không bị load lại.
  - Ví dụ: youtube.com, facebook.com, gmail.com,...



## 3.6. Template Engines



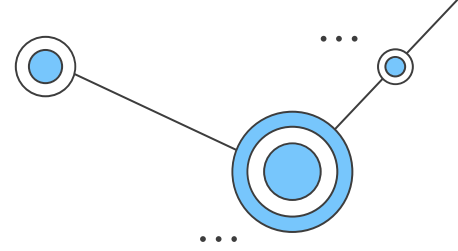
- Khái niệm
  - Template Engines là công cụ giúp tách mã HTML thành các phần nhỏ hơn.
  - Để tái sử dụng lại trên nhiều file HTML.
- Một số template engines phổ biến
  - **PUG**
  - EJS
  - Handlebars
- Sử dụng Pug:
  - Link cài đặt: <https://www.npmjs.com/package/pug>
  - Link tài liệu: <https://pugjs.org/api/getting-started.html>





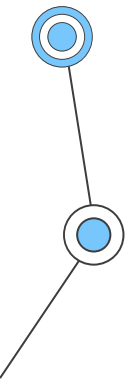
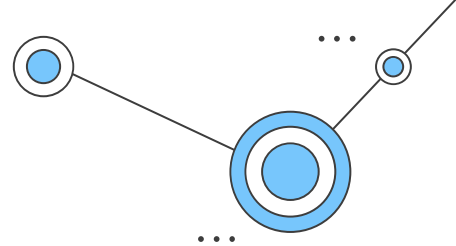
## 3.7. Static files & CSS

- *Hướng dẫn trong lúc học.*



## 3.8. Sử dụng Bootstrap 4

- Link trang chủ: <https://getbootstrap.com/>
- *Hướng dẫn trong lúc học.*





04

PUG



## 4. PUG

- **Doctype** (Kiểu tài liệu)

Cú pháp PUG

```
doctype html
```

Biên dịch thành HTML

```
<!DOCTYPE html>
```

## 4. PUG

- **Tags** (Thẻ)

### Cú pháp PUG

```
h1 Nội dung thẻ h1
h2 Nội dung thẻ h2
p Nội dung thẻ p
img
```

### Biên dịch thành HTML

```
<h1>Nội dung thẻ h1</h1>
<h2>Nội dung thẻ h2</h2>
<p>Nội dung thẻ p</p>
<img />
```

```
ul
  li Item 1
  li Item 2
  li Item 3
```

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

## 4. PUG

- **Plain Text** (Văn bản)

Cú pháp PUG

p Nội dung thẻ p!

p Nội dung **<b>**thẻ p</b>!

p Nội dung *<i>*thẻ p</i>!

p Nội dung **<b><i>**thẻ p</i></b>!

Biên dịch thành HTML

<p>Nội dung thẻ p!</p>

<p>Nội dung **<b>**thẻ p</b>!</p>

<p>Nội dung *<i>*thẻ p</i>!</p>

<p>Nội dung **<b><i>**thẻ p</i></b>!</p>

## 4. PUG

- **Plain Text** (Văn bản)

### Cú pháp PUG

```
p
| Lorem ipsum dolor sit amet
| consectetur, adipisicing elit.
```

### Biên dịch thành HTML

```
<p>Lorem ipsum dolor sit amet
consectetur, adipisicing elit.</p>
```

```
p.
  Lorem ipsum dolor sit, amet
  consectetur adipisicing elit.
  Harum ipsam voluptatum suscipit
  labore temporibus?
```

```
<p>Lorem ipsum dolor sit, amet
consectetur adipisicing elit. Harum
ipsam voluptatum suscipit labore
temporibus?</p>
```

## 4. PUG

- **Attributes** (Thuộc tính)

### Cú pháp PUG

```
button.btn.btn-success#button-test Click Me 1
```

```
button(class="btn btn-success", id="button-test") Click Me 2
```

```
button(class="btn btn-success" id="button-test") Click Me 3
```



```
<button class="btn btn-success" id="button-test">Click Me 1</button>
```

```
<button class="btn btn-success" id="button-test">Click Me 2</button>
```

```
<button class="btn btn-success" id="button-test">Click Me 3</button>
```

Biên dịch thành HTML



## 4. PUG

- **Comments** (Chú thích)

Cú pháp PUG

```
// Đây là đoạn comment được biên dịch thành mã HTML  
//- Đây là đoạn comment không được biên dịch thành mã HTML
```

<!-- Đây là đoạn comment được biên dịch thành mã HTML-->

Biên dịch thành HTML

## 4. PUG

- **Variable** (Biến)

### Cú pháp PUG

```
- var name = "Đặng Phương Nam";  
- var phone = "0123456789";  
- var age = 18;
```

```
div(class="info")  
  p= name  
  p= phone  
  p= age
```

```
div(class="info")  
  p #{name}  
  p #{phone}  
  p #{age}
```

```
div(class="info")  
  p Họ tên: #{name}  
  p Điện thoại: #{phone}  
  p Tuổi: #{age}
```

### Biên dịch thành HTML

```
<div class="info">  
  <p>Đặng Phương Nam</p>  
  <p>0123456789</p>  
  <p>18</p>  
</div>
```

```
<div class="info">  
  <p>Đặng Phương Nam</p>  
  <p>0123456789</p>  
  <p>18</p>  
</div>
```

```
<div class="info">  
  <p>Họ tên: Đặng Phương Nam</p>  
  <p>Điện thoại: 0123456789</p>  
  <p>Tuổi: 18</p>  
</div>
```

## 4. PUG

- **Variable** (Biến)

### Cú pháp PUG

```
- var htmlLink = "<a href='https://daca.vn/'>Daca.vn</a>";  
  
div(class="link")  
  p= htmlLink  
  
div(class="link")  
  p #{htmlLink}  
  
div(class="link")  
  p !{htmlLink}
```

↓

```
<div class="link"><p>&lt;a href='https://daca.vn/'&gt;Daca.vn&lt;/a&gt;</p></div>  
<div class="link"><p>&lt;a href='https://daca.vn/'&gt;Daca.vn&lt;/a&gt;</p></div>  
<div class="link"><p><a href='https://daca.vn/'>Daca.vn</a></p></div>
```

Biên dịch thành HTML

## 4. PUG

- **Conditionals** (Điều kiện)

Cú pháp PUG

```
-  
var infoUser = {  
  name: "Đặng Phương Nam",  
  phone: "0123456789",  
  age: 18  
};  
  
div(class="info")  
  p Email: #{infoUser.email}  
  
if infoUser.email  
  p Email: #{infoUser.email}
```

Biên dịch thành HTML

```
<div class="info">  
  <p>Email: </p>  
</div>
```

## 4. PUG

- **Case** (Trường hợp)

### Cú pháp PUG

```
//- statusAccount: "initial", "active", "inactive"  
- var statusAccount = "initial";  
  
case statusAccount  
  when "initial"  
    p Chờ duyệt  
  when "active"  
    p Đang hoạt động  
  when "inactive"  
    p Đang bị khóa  
  default  
    p Không có
```

### Biên dịch thành HTML

```
<p>Chờ duyệt</p>
```

## 4. PUG

- **Iteration** (Vòng lặp)

Cú pháp PUG

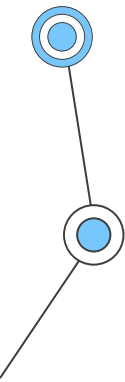
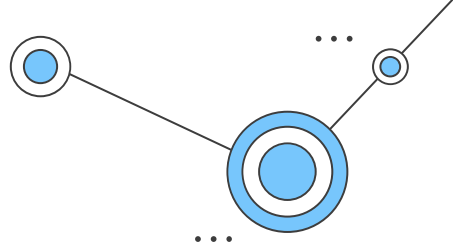
```
-  
var listUser = [  
  "Le Van A",  
  "Nguyen Thi B",  
  "Do Van C"  
];  
  
ul  
  each item, index in listUser  
    li #{index+1}. #{item}
```

Biên dịch thành HTML

```
<ul>  
  <li>1. Le Van A</li>  
  <li>2. Nguyen Thi B</li>  
  <li>3. Do Van C</li>  
</ul>
```

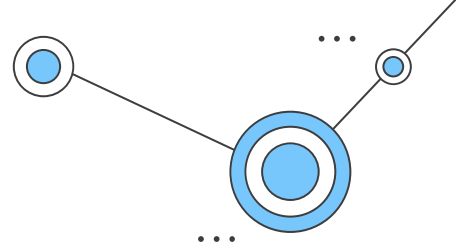
## 4. PUG

- **Template Inheritance** (Kế thừa mẫu)
  - Dùng để tạo ra một bộ khung dùng chung được ở nhiều trang.
  - *Hướng dẫn trong lúc học.*



## 4. PUG

- **Includes**
  - Dùng để nhúng nội dung của 1 file PUG vào 1 file PUG khác.
  - *Hướng dẫn trong lúc học.*





## 4. PUG

- **Mixins**
  - Dùng để tạo ra các khối có thể tái sử dụng.
  - *Hướng dẫn trong lúc học.*

