**NODEJS**

# 1.Khái niệm nodejs? Là môi trường thực thi (platform) javascript ở server-side

Node.js là một nền tảng phát triển server-side (phía máy chủ) mã nguồn mở, được xây dựng trên nền tảng V8 JavaScript engine của Google. Nó cho phép các nhà phát triển sử dụng JavaScript để phát triển các ứng dụng web phía server.

* Node.js cung cấp một số tính năng đáng chú ý như khả năng xử lý các yêu cầu đồng thời.
* Tích hợp với các thư viện và framework phổ biến để phát triển các ứng dụng web như Express.js và Socket.io.
* Hỗ trợ một loạt các giao thức mạng như HTTP, TCP và UDP.
* Node.js cho phép các nhà phát triển tạo ra các ứng dụng web phía server đơn giản và hiệu quả, cũng như giải quyết các vấn đề về hiệu suất và khả năng mở rộng trong phát triển ứng dụng web.
* Node.js cũng có thể được sử dụng để phát triển các ứng dụng desktop và mobile thông qua các framework như Electron và React Native.
* Node.js là một công nghệ phổ biến trong cộng đồng lập trình và được sử dụng rộng rãi trong các dự án phát triển ứng dụng web.

# 2. Engine V8 trong Node.js?

* Engine V8 trong Node.js là một trình biên dịch mã nguồn mở được phát triển bởi Google, được sử dụng để biên dịch mã JavaScript thành mã máy để chạy trên nền tảng Node.js.
* Engine V8 trong Node.js cung cấp hiệu suất tốt hơn cho các ứng dụng JavaScript, cho phép Node.js chạy các ứng dụng phức tạp và xử lý lượng dữ liệu lớn một cách nhanh chóng và hiệu quả hơn.
* Engine V8 sử dụng công nghệ bộ nhớ động để quản lý bộ nhớ và thu thập rác tự động, giúp giảm thiểu sự cố về quản lý bộ nhớ.

Ngoài ra, Engine V8 trong Node.js cũng cung cấp một loạt các tính năng bảo mật như sandboxing và tường lửa để bảo vệ các ứng dụng khỏi các lỗ hổng bảo mật tiềm ẩn.

# 3. Express js là gì?

Express.js là một framework phổ biến cho Node.js, được sử dụng để xây dựng các ứng dụng web và API RESTful.

Nó cung cấp một cách tiếp cận đơn giản và linh hoạt để xử lý các yêu cầu HTTP và phản hồi tương ứng.

Express.js có thể được sử dụng để xây dựng các ứng dụng web đơn giản hoặc phức tạp, cho phép phát triển theo hướng tùy chỉnh và đáp ứng các yêu cầu của dự án.

Một số tính năng nổi bật của Express.js bao gồm:

* Xử lý các yêu cầu HTTP và phản hồi tương ứng dễ dàng
* Hỗ trợ các phương thức HTTP như GET, POST, PUT, DELETE, vv.
* Hỗ trợ middleware để thực hiện các chức năng như xác thực, bảo mật, ghi log, vv.
* Hỗ trợ routing để xử lý các yêu cầu cho các URL cụ thể
* Hỗ trợ template engine để dễ dàng hiển thị các nội dung động cho người dùng
* Với Express.js, các nhà phát triển có thể tập trung vào việc xây dựng các chức năng cốt lõi của ứng dụng mà không cần quan tâm đến các thủ tục tầm thấp như xử lý yêu cầu và phản hồi.

# 4. Khái niệm client và server là gì?

Client và server là hai khái niệm cơ bản trong mô hình client-server, là một kiểu kiến trúc phần mềm phổ biến được sử dụng trong hệ thống mạng và ứng dụng web.

* Client: là một chương trình hoặc thiết bị sử dụng dịch vụ của một máy chủ. Client thực hiện yêu cầu gửi đến server và chờ đợi server trả về kết quả để xử lý. Ví dụ như trình duyệt web, phần mềm email, ứng dụng đọc tin nhắn,....
* Server: là một máy tính hoặc thiết bị phần cứng chạy các dịch vụ, ứng dụng và cung cấp các tài nguyên để phục vụ các yêu cầu từ client. Ví dụ như máy chủ web, máy chủ email, máy chủ đa phương tiện,....

Mối quan hệ giữa client và server là bằng cách client gửi các yêu cầu đến server và đợi server phản hồi với kết quả. Các giao tiếp này thường được thực hiện thông qua mạng internet hoặc mạng nội bộ của tổ chức. Quan hệ giữa client và server đóng vai trò quan trọng trong việc xây dựng các ứng dụng web và mạng, cung cấp tính linh hoạt, tiện lợi và tăng hiệu suất cho người sử dụng.

# 5.NPM (NODE PACKAGE MANAGER)

* **Tạo file package.json:** npm init
* **Tạo express:** npm install --save-exact express@4.17.1
* **Cài đặt view engine EJS:** npm install --save-exact [ejs@3.1.6](mailto:ejs@3.1.6)
* **Sử dụng việc phân tích và xử lý các yêu cầu HTTP:** npm install --save-exact [body-parser@1.19.0](mailto:body-parser@1.19.0)
* **Restart server:** npm install --save-exact [nodemon@2.0.12](mailto:nodemon@2.0.12)
* **Cài đặt babel:** npm install @babel/core @babel/node @babel/preset-env
* **@babel/core** là package chính của Babel, cung cấp các API để phân tích cú pháp (parse), biên dịch (transform) và tạo lại mã nguồn (generate) trong quá trình transpiling mã ECMAScript mới (ES6, ES7,...) thành mã JavaScript tương thích với phiên bản trình duyệt cũ hơn:
* npm install --save-exact [@babel/core@7.15.](mailto:nodemon@2.0.12)5
* **@babel/node** là một package giúp thực thi mã ECMAScript mới trên Node.js với trình thông dịch (interpreter) của Babel. Điều này cho phép chúng ta viết mã ECMAScript mới trực tiếp trên Node.js mà không cần phải biên dịch trước đó.
* npm install --save-exact [@babel/node@7.15.](mailto:nodemon@2.0.12)4
* **@babel/preset-env** là một preset của Babel, nó giúp định nghĩa các cài đặt để biên dịch mã ECMAScript mới thành phiên bản JavaScript tương thích với phiên bản trình duyệt hoặc môi trường mà chúng ta đang sử dụng. Preset này cho phép chúng ta chỉ định các tính năng ECMAScript mới mà chúng ta muốn sử dụng, và Babel sẽ biên dịch các tính năng đó thành phiên bản tương thích với môi trường hiện tại.
* npm install --save-exact [@babel/preset-env@7.15.](mailto:nodemon@2.0.12)6

npm install --save-exact [body-parser@1.19.0](mailto:body-parser@1.19.0) [nodemon@2.0.12](mailto:nodemon@2.0.12) [@babel/core@7.15.](mailto:nodemon@2.0.12)5 [@babel/node@7.15.](mailto:nodemon@2.0.12)4 [@babel/preset-env@7.15.](mailto:nodemon@2.0.12)6

* **Cài đặt môi trường để chạy đc các file .env:** npm install --save-exact [dotenv@10.0.0](mailto:dotenv@10.0.0)
* **Sử dụng mySQL2:** npm install --save-exact [mysql2@2.3.0](mailto:mysql2@2.3.0)
* **Cài đặt sequelize:** npm install --save-dev [sequelize-cli@6.2.0](mailto:sequelize-cli@6.2.0)

**Cài đặt sequelize:** npm install --save sequelize@6.6.2

* **Cài đặt config sequelize:** npx sequelize-cli init
* **Kết nối DB:** npm install --save [mysql2@2.2.5](mailto:mysql2@2.2.5)
* **Tạo model:** npx sequelize-cli model:generate --name User --attributes firstName:string,lastName:string,email:string
* **Tại thư mục root, sử dụng câu lệnh:** node\_modules/.bin/sequelize init
* **Tạo migrations:** npx sequelize-cli db:migrate
* **Tạo Seeder:** npx sequelize-cli seed:generate --name demo-user

# 6.Sử dụng express?

const express = require('express')

const app = express()

const port = 3000

app.get('/', (req, res) => {

  res.send('Hello World!Ajinomoto')

})

app.listen(port, () => {

  console.log(`Example app listening on port ${port}`)

})

* Cung cấp một máy chủ web địa phương trên cổng 3000 và trả về chuỗi 'Hello World!Ajinomoto' khi một yêu cầu GET được thực hiện tới đường dẫn cơ bản '/'.
* Dòng đầu tiên, const express = require('express'), là để tải Express framework.
* Tiếp theo, const app = express() tạo ra một ứng dụng Express mới và lưu nó vào biến app.
* Dòng const port = 3000 xác định cổng mà ứng dụng sẽ lắng nghe.
* app.get('/', ...) là phương thức đăng ký xử lý cho yêu cầu GET đến đường dẫn cơ bản của ứng dụng. Khi yêu cầu GET đến '/', res.send() sẽ gửi lại phản hồi với nội dung là chuỗi "Hello World!Ajinomoto".
* Cuối cùng, app.listen(port, ...) lắng nghe các yêu cầu đến cổng đã được xác định và ghi ra console khi máy chủ web đã sẵn sàng để nhận các kết nối đến.

# 7. VIEW ENGINE trong nodejs?

**"View engine"** là một thành phần quan trọng của Node.js framework như Express, được sử dụng để hiển thị các trang web động hoặc các giao diện người dùng khác.

**View engine** giúp bạn tạo các template cho trang web của mình và thực hiện việc kết hợp dữ liệu và template đó để hiển thị cho người dùng.

**View engine:** xử lý các điều kiện logic bên trong trang HTML.

Một số view engine phổ biến trong Node.js là:

* EJS (Embedded JavaScript) - cho phép nhúng mã JavaScript vào các template của bạn và cung cấp các hàm tiện ích để giúp bạn xử lý dữ liệu => thường sử dụng làm view engine cho nodejs, cú pháp ngắn gọn dễ tiếp cận.
* Pug (trước đây là Jade) - là một ngôn ngữ template đơn giản nhưng mạnh mẽ, cung cấp một cú pháp ngắn gọn và dễ đọc để xây dựng các trang web.
* Handlebars - cho phép bạn tạo ra các template động mà có thể được sử dụng trên cả phía máy chủ và phía máy khách….
* …

Việc lựa chọn view engine phụ thuộc vào sở thích và nhu cầu của bạn, và tất cả các view engine đều có những ưu điểm và nhược điểm riêng.

* Link EJS github: <https://github.com/mde/ejs>

# 8. body-parser trong js là gì?

* body-parser là một middleware trong Node.js được sử dụng để lấy dữ liệu từ phần thân (body) của một request được gửi đến máy chủ (server) thông qua HTTP request.
* Khi một request HTTP được gửi đến server, thông tin của request đó được gửi dưới dạng header và body. Trong phần thân của request, chúng ta có thể gửi dữ liệu dưới nhiều định dạng khác nhau, ví dụ như JSON, form data, hoặc text.
* body-parser được sử dụng để phân tích và lấy dữ liệu từ phần thân của request và đưa vào biến request.body. Điều này giúp cho chúng ta có thể dễ dàng truy cập và xử lý các dữ liệu này trong ứng dụng Node.js của mình.

# 9. static file có ý nghĩa gì?

1. **Static file** là các tệp tin dữ liệu như hình ảnh, CSS, JavaScript, font chữ,... được sử dụng để hiển thị giao diện và các chức năng tĩnh của một trang web. Những tệp tin này không thay đổi nội dung dữ liệu của trang web khi người dùng truy cập vào, nghĩa là chúng không được tạo ra theo yêu cầu của người dùng mà được tải lên trước và lưu trữ trên máy chủ.
2. Static file là một phần quan trọng của một trang web, vì chúng giúp giảm thiểu thời gian tải trang web và cải thiện trải nghiệm người dùng. Khi trình duyệt tải trang web, nó sẽ tải các tệp tin dữ liệu tĩnh trước khi hiển thị trang web cho người dùng, giúp cho trang web được tải nhanh hơn và tiết kiệm băng thông Internet của người dùng.

# 10. Khái niệm ROUTERS trong nodejs?

Trong Node.js, Router là một thành phần của framework Express.js, được sử dụng để xác định các đường dẫn và xử lý các yêu cầu HTTP tương ứng.

Router được sử dụng để phân tách mã xử lý yêu cầu từ các tệp chính của ứng dụng, giúp giảm thiểu sự phức tạp của mã và dễ dàng bảo trì ứng dụng. Router cung cấp một cách để xác định các đường dẫn và các phương thức HTTP khác nhau, chẳng hạn như GET, POST, PUT và DELETE, để xử lý các yêu cầu tương ứng.

# 11. npm là gì?

**npm** là viết tắt của **Node Package Manager**, là một công cụ quản lý gói phần mềm cho Node.js. npm cho phép người dùng cài đặt, quản lý và chia sẻ các thư viện, module, ứng dụng Node.js cũng như các công cụ liên quan.

**npm** được cài đặt sẵn khi cài đặt Node.js và có thể được sử dụng để tải xuống các gói phần mềm Node.js từ kho lưu trữ npm, hoặc tạo và quản lý các gói phần mềm của riêng bạn.

**npm** cũng cung cấp cho người dùng một loạt các tính năng quản lý phụ thuộc, tạo các tệp tóm tắt danh sách các phụ thuộc và phiên bản của chúng trong dự án, và thực hiện các lệnh để cập nhật hoặc gỡ bỏ các phụ thuộc.

Bottom of Form

Top of Form