

GIỚI THIỆU VÀ TỔNG QUAN VỀ VUE JS

- NGUYỄN VĂN MẠNH -



MỤC TIÊU BÀI HỌC

- Hiểu được tổng quan về Vue
- Những tính năng của Vue
- Những tính năng hữu ích nhất của Vue.
- Những khả năng và hạn chế của Vue.
- So sánh với **Jquery**, Angular và React
- Các thư viện thú vị của hệ sinh thái vue
- Cài đặt, tạo một ứng dụng Vue đơn giản



"Some are born great, some achieve greatness, and some have greatness thrust upon'em"

- William Shakespeare, Twelfth Night -

VUE LÀ GÌ?

- Vue là một thư viện của Javascript. Một khối mã lớn js được chạy trước bất kỳ đoạn mã nào bạn viết và thực thi.
- Là một công cụ kết xuất (render engine). Lấy mô hình dữ liệu (data model) và các thành phần (components) làm đầu vào và xuất ra HTML - thứ mà trình duyệt có thể hiểu và biến thành thứ mà người dùng có thể nhìn thấy được.
- Là một công cụ cộng tác (collaboration tool), cho phép chia ứng dụng thành các phân đoạn nhỏ hơn để nhiều người làm việc trên chúng thông qua các components.

CÁC TÍNH NĂNG CỐT LÕI LÀ GÌ?

- Vue có các thành phần linh hoạt (versatile components)
- Vue cũng có khả năng “phản ứng” tích hợp sẵn (**build-in reactivity**).
- Tốc độ tải nhanh, Vue được thiết kế để hiển thị nhanh cho UI.
- Vue đi kèm với những chỉ thị quen thuộc (familiar directives) như for, if, bind (v-for, v-if, v-bind) như React hoặc Angular hoặc thậm trí là cả Knockout.

TỆP .VUE - ĐẶC THÙ CỦA VUE?

- Tập đặc thù của Vue được gọi là **vue file**.
- .vue files chứa HTML, CSS và các đoạn mã javascript
- Các file .vue không thể đọc được bằng trình duyệt.
- Các tệp .vue cho phép gộp tất cả mã trong một component đơn thành 1 file duy nhất và không cần viết bất kỳ một đoạn mã nào (kể cả html và javasctip).

NHỮNG LỢI ÍCH CỦA VUE?

- Tốc độ tạo, render trang nhanh hơn nhiều so với js nguyên bản.
- Đặt được nhiều loại logic khác nhau tập trung trong tệp .vue -> có ít tệp hơn.
- Khả năng tương tác đơn giản, dễ dàng có được các thành phần của react JQuery hoặc bất cứ framework nào khác mà nó không có sẵn, tận dụng được các triết lý lập trình như Angular đủ đơn giản để có thể làm việc với Vue framework.
- Vue dễ học.

SO SÁNH VỚI CÁC FRAMEWORK KHÁC


Vue.js	Jquery, React, Angular...
Lập trình ngay lập tức với Vue nguyên mẫu.	Tìm ra cấu trúc thư mục, công cụ sử dụng, cách thức build và thực hiện từng bước đó để tạo 1 ứng dụng mới. Có thể sử dụng CLI nhưng có rất nhiều phụ thuộc
Đặt tất cả các logic, style và tất cả các cấu trúc khác (VD Docker) trong 1 file .vue	Logic của ứng dụng được đặt trong các controller files: js, css hoặc ít nhất là đặt trong các tệp JSX hoặc HTML. Sau đó sử dụng import để gộp lại thành 1 tệp. (Chỉ phù hợp nếu muốn làm việc chuyên biệt với css/js..)
Dễ học, dễ tiếp cận, không tốn nhiều chi phí về thời gian.	Phải học nhiều thứ từ công cụ, cách build... mất nhiều thời gian hơn.

HẠN CHẾ?

- Quá linh hoạt nên sẽ có quá nhiều lựa chọn cho các lập trình viên. Khó tìm được một chuẩn mã nhất quán cho toàn bộ nhóm.
- Tài liệu chuẩn tiếng anh hiện chưa tốt và hoàn thiện như tiếng Trung (do lượng cộng đồng sử dụng lớn có lượng người TQ nhiều). Tài liệu chưa hoàn thiện như Angular hay React.
- Thiếu lập trình viên Vue làm việc toàn thời gian. Thường chỉ làm part-time.

WHY VUE.JS

CONSIDERED THE BEST FRAMEWORK FOR FRONT-END
DEVELOPMENT FOR 2017

The Vue.js logo is a large, stylized letter 'V' in the center. It is composed of two main colors: a dark blue/black for the central negative space and a vibrant green for the outer parts. The 'V' is flanked by two sets of horizontal, slightly overlapping bars that create a sense of motion or wings. The entire logo is set against a light gray background with a subtle gradient.

TẠI SAO LẠI SỬ DỤNG VUE?

VUE CÓ THỂ LÀM GÌ? KHÔNG THỂ LÀM GÌ?

Vue có thể làm gì?	Vue không làm được gì?
Vue xuất ra các mã HTML cho trình duyệt.	Không tạo ra một server cho ứng dụng (Trường hợp ngoại lệ nếu sử dụng Vue CLI vì Vue CLI có thể tạo ra một server nhưng nó chạy trên npm).
Vue cũng giống các UI Framework khác: có thể map các tương tác của người dùng với các phương thức js mà bạn viết (VD click, hover...).	Vue không kiểm tra mã để tìm lỗi, ta phải tự làm việc này. Nếu viết sai mã thì Vue vẫn cố gắng chạy nó, không giống như Angular, nó không có nhiều các phương thức tiện ích.
Sử dụng các kỹ thuật phức tạp như React và Angular để tăng tốc độ render của ứng dụng.	Không có định tuyến URL routing (cần phải sử dụng thêm một thư viện là Vue-router)
Vue hoạt động trên mọi trình duyệt mà không cần thực hiện bất cứ một bước build (dựng) nào cả.	Không cung cấp các giải pháp riêng cho chính nó ngoài các giải pháp gốc có sẵn của ES6.

REACTIVITY VÀ VUE

- **Reactivity** là một trong những vấn đề chính mà Vue đặt ra để giải quyết.
- **Reactivity là gì?:** là cách mà ứng dụng thay đổi những gì người dùng thấy (UI) để phản hồi lại những hành động của người (VD người dùng nhấn vào một button, nhập liệu vào textbox...)

TẠI SAO REACTIVITY LẠI QUAN TRỌNG?

- JS thuần/ JQuery không có một cơ chế tích hợp bất kỳ nào để làm được việc như **Reactivity**. Cơ chế này của React, Angular, Vue đã lấp đầy khoảng trống đó.
- Nếu được triển khai không đúng cách có thể sẽ **Reactivity** sẽ rất chậm, nó có thể sẽ là nút thắt làm chậm ứng dụng. (Vue đã giải quyết được vấn đề này, không giống như cách của Angular hay React – vue làm tốt hơn).

VUE ĐÃ XỬ LÝ REACTIVITY NHƯ THẾ NÀO?

- Vue **tự động** phát hiện những thay đổi của dữ liệu.
- Vue **tự động** cập nhật những gì mà người dùng nhìn thấy và người dùng có thể tiếp tục tương tác với ứng dụng ngay lập tức.
- Vue **tái sử dụng lại các phần tử DOM** và cũng sử dụng các kỹ thuật đặc biệt khác để nâng cao hiệu suất.

REACTIVITY CỦA VUE CÓ Ý NGHĨA NHƯ THẾ NÀO VỚI LẬP TRÌNH VIÊN VÀ NGƯỜI DÙNG?

- **Đối với LTV:** mã viết dễ hơn, ít hơn, dễ bảo trì hơn vì bạn không phải viết các đoạn mã cụ thể cho việc muốn hiển thị một thông tin cho người dùng. Vue **sử dụng một DOM ảo** để thực hiện cập nhật nhanh chóng.
- **Đối với người dùng cuối:** ứng dụng nhanh và cảm thấy thú vị khi sử dụng. Kể cả người dùng có làm việc với một danh sách dữ liệu rất lớn thì Vue là một lựa chọn số 1, vì hầu như tốc độ không bị ảnh hưởng bởi lượng dữ liệu. Điều này sẽ khiến người dùng có trải nghiệm tốt.

NHỮNG THÁCH THỨC CỦA REACTIVITY VÀ VUE?

- Vue đang **cố gắng làm mọi thứ một cách tự động** dẫn đến một vài tình huống phát sinh sẽ khó giải quyết. Các Vue components được cập nhật tự động khi dữ liệu thay đổi. Nó sẽ hữu ích và rất đơn giản nếu dữ liệu là một số, chuỗi, boolean hoặc một cấu trúc dữ liệu bất biến. Không như mảng và đối tượng (kiểm soát chưa tốt).
- Vue không thể luôn luôn biết chính xác là mảng đã được cập nhật hay chưa?
- Với các loại mutable data (dạng dữ liệu thay đổi như array hay object), view có thể sẽ không cập nhật một cách chính xác.

HỆ SINH THÁI VUE (VUE ECOSYSTEM) CÁC THƯ VIỆN HAY HO



1. Nuxt (UNIVERSAL RENDERING)

- Sử dụng để tạo một ứng dụng Vue phổ quát (do máy chủ kết xuất).
- Nuxt giải quyết rất nhiều những thách thức phức tạp về kỹ thuật và kiến thức cả về front-end và back-end (mà bạn cần biết) khi bạn tạo mới một ứng dụng với Vue.
- Thời gian load rất nhanh trên tất cả các thiết bị, đặc biệt là điện thoại di động.

2. VUEX (STATE MANAGEMENT)

- Cung cấp cách thức cập nhật trạng thái (state) cho các component. Bản chất trạng thái sẽ ảnh hưởng đến các thành phần mà bạn sẽ xem xét và tương tác của người dùng có thể làm thay đổi trạng thái đó (như là thêm mới hoặc cập nhật thông tin cá nhân).
- **Vuex** giống như **Flux** hoặc **Redux**.
- Sử dụng để quản lý ứng dụng có lượng state lớn (VD ứng dụng có hàng trăm hoặc hàng ngàn thông tin khác nhau).

3. VUETIFY (COMPONENT FRAMEWORK)

- Chứa nhiều các component mượt mà và đẹp. LTV sẽ không mất quá nhiều công sức để viết. (Như Bootstrap với HTML)
- Các component này đều có thiết kế rất đẹp mắt. Đặc biệt là cho mobile.
- Được sử dụng như một giải pháp thay thế cho việc tự phải tạo ra giải pháp riêng cho từng thành phần của UI. Tiết kiệm được thời gian, tiền bạc...

4. VUE-CLI (SCAFFOLDING & BUILD)

- Có vai trò rất quan trọng. Là công cụ để tạo, dựng (build) và cập nhật (updating) ứng dụng Vue.
- Vue-CLI cung cấp cách thức tiêu chuẩn để cập nhật ứng dụng cho toàn dự án. Không cần những LTV cấp cao, chuyên nghiệp mới có thể làm các công việc này.
- Cho phép những lập trình viên mới dễ dàng thực hiện làm việc với mã nguồn ứng dụng (VD: Xây dựng thêm các tính năng mới).
- Nâng cao năng suất với một đội ngũ dự án lớn - càng lớn càng có ý nghĩa (không thực sự cần thiết cho 1 dự án nhỏ 1-2 người).



ĐỂ LÀM VIỆC VỚI VUE THÌ CẦN NHỮNG GÌ?

Để làm việc với vue thì cần những gì?

- Kiến thức cơ bản về HTML/CSS/JS
- Biết sử dụng một công cụ soạn thảo văn bản (khuyến dùng Atom/WebStorm/ **VS Code**).
- Có một máy chủ cục bộ (local server): Apache/ Http-server/ Vue CL

VUE – GETTING STARTED - DEMO



TOOLING

- Cài đặt **NodeJs**
- Cài đặt hệ điều hành LTS (Long Time Support)
- Cài đặt **Vue**:
`npm install -g @vue-cli`
- **Visual Studio Code**
- **Vetur** and **Vue Snippets** extensions for VS Code.
- Vue DevTool Extend cho Chrome để thực hiện Debug.

DEMO

- Tạo mới App với Vue:

```
vue create [app name] -d
```

- Chuyển thư mục:

```
cd [folder]
```

- Server app:

```
npm run serve
```

