



LẬP TRÌNH REACT THẬT ĐƠN GIẢN

Từng bước tạo ứng dụng từ A-Z

By VNTALKING



MỤC LỤC

| | |
|---|-----------|
| LỜI NÓI ĐẦU | 4 |
| NỘI DUNG CUỐN SÁCH..... | 5 |
| AI NÊN ĐỌC CUỐN SÁCH NÀY? | 6 |
| Yêu cầu trình độ | 6 |
| Cách học đúng cách | 6 |
| Liên hệ tác giả | 7 |
| GIỚI THIỆU | 8 |
| Ưu điểm của React..... | 8 |
| TỔNG QUAN REACT..... | 11 |
| Virtual DOM | 11 |
| DOM là gì? | 11 |
| Vấn đề của DOM là gì? | 11 |
| Cơ chế hoạt động của Virtual DOM | 12 |
| CÀI ĐẶT REACT | 13 |
| Cách 1: Viết React trực tiếp trong HTML | 13 |
| Cách 2: Sử dụng Create-React-App CLI..... | 17 |
| Cài đặt Code Editor và các extension cần thiết | 20 |
| XÂY DỰNG ỨNG DỤNG TODOS | 26 |
| Giới thiệu ứng dụng Todos | 26 |
| Những kiến thức React cơ bản nhất | 27 |
| Tạo các components..... | 32 |
| Tạo Header component và thêm Styles cho ứng dụng Todo | 37 |
| Thêm State | 39 |
| Tạo component hiển thị danh sách Todos | 42 |
| Controlled Component | 46 |
| Gửi và xử lý sự kiện - Handle Events..... | 48 |
| Cập nhật giá trị state sử dụng hàm setState()..... | 51 |
| Sử dụng Destructuring..... | 53 |

| | |
|---|------------|
| Xóa một Todo..... | 54 |
| Thêm một Todo mới..... | 56 |
| Cập nhật Todo mới vào danh sách..... | 60 |
| FETCHING DỮ LIỆU TỪ API | 64 |
| Lý thuyết cơ bản về Request Network | 64 |
| Giới thiệu API cung cấp cho ứng dụng Todo | 66 |
| Vòng đời Component trong React | 67 |
| Cài đặt axios..... | 68 |
| Lấy danh sách Todos bằng GET request | 69 |
| Thêm Todo bằng POST request | 71 |
| Xóa một Todo bằng DELETE request | 72 |
| REACT HOOKS..... | 74 |
| Tìm hiểu Hooks | 74 |
| Refactoring mã nguồn Todo App sử dụng Hooks | 78 |
| QUẢN LÝ STATE VỚI REDUX..... | 83 |
| 3 nguyên tắc của Redux | 83 |
| Khi nào thì sử dụng Redux?..... | 83 |
| Thành phần của Redux | 84 |
| Thực hành sử dụng Redux trong Todo App | 85 |
| DEPLOY ỨNG DỤNG REACT | 94 |
| Deploy ứng dụng React lên Github Pages..... | 95 |
| BÀI TẬP..... | 100 |
| Project task | 101 |
| Đáp án bài tập tham khảo | 103 |
| KẾT NỐI VỚI VNTALKING | 105 |
| THÔNG TIN VỀ TÁC GIẢ..... | 106 |
| CUỐN SÁCH KHÁC CỦA VNTALKING | 107 |

LỜI NÓI ĐẦU

Nói đến front-end là không thể không nhắc tới ReactJS. Một thư viện Javascript được chống lưng bởi ông lớn Facebook. Lướt qua các trang tuyển dụng lớn, bạn sẽ thấy nhu cầu tuyển người biết ReactJS rất lớn.

Bạn biết CMS nổi tiếng Wordpress không? Từ Wordpress 5.0, hẳn bạn đã nghe tới trình soạn thảo Gutenberg được tích hợp sẵn, nó cũng được xây dựng bằng React đó.

Thêm một ví dụ nữa, đó là Gatsby, một trình tạo trang web tĩnh nổi tiếng, một xu hướng mà nhiều nhà phát triển bắt đầu chuyển sang.

Và rất nhiều ứng dụng nổi tiếng khác cũng đang sử dụng React như: chotot, Shopee, Lazada, MyTV...

Điều này cho thấy rằng, việc lựa chọn ReactJS là một lựa chọn thông minh trong thời điểm hiện tại.

Với mục đích xây dựng nền tảng khi bắt đầu học ReactJS, mình cho ra đời cuốn sách này.

Trong cuốn sách này, bạn sẽ không chỉ học các nguyên tắc cơ bản của ReactJS, mà bạn còn có thể ứng dụng chúng để tự xây dựng các ứng dụng ReactJS hiện đại và triển khai miễn phí lên Internet.

Ngoài ra, trong cuốn sách, mình sẽ giới thiệu những kỹ thuật hỗ trợ bạn xây dựng ứng dụng ReactJS dễ dàng hơn. Ví dụ như quản lý mã nguồn với GIT, lựa chọn và cài đặt Code Editor phù hợp với ReactJS.v.v...

Nếu bạn là người thích sự chi tiết và tỉ mỉ thì cuốn sách này đích thị dành cho bạn.

Hãy tiếp tục đọc và nghiền ngẫm, bạn sẽ cảm thấy yêu thích cuốn sách này.

Mình đảm bảo!

NỘI DUNG CUỐN SÁCH

Chào mừng bạn đến với cuốn sách: "**Lập trình React thật đơn giản**". Mình tin chắc rằng, bạn được nghe không nhiều thì ít về ngôn ngữ lập trình web như HTML, Javascript, CSS.

Về cơ bản, để tạo một ứng dụng web (ám chỉ front-end), bạn chỉ cần biết 3 thứ: HTML, Javascript, CSS. Tất cả những công nghệ, thư viện, frameworks... dù đao to búa lớn thì cũng chỉ xoay quanh 3 ngôn ngữ đó mà thôi.

Thế giới web hiện đại ngày nay, chúng ta không chỉ đơn thuần xây dựng các website cung cấp thông tin một chiều mà còn có các ứng dụng web, khi mà người dùng có thể tương tác, thay đổi nội dung, thay đổi giao diện mà không phải tải lại trang (chính là các ứng dụng web dạng Single Page Applications - gọi tắt là SPA).

Và đây là mảnh đất để các thư viện như ReactJS tung hoành.

Trong cuốn sách này, bạn sẽ được học và thực hành:

- Cách cài đặt và bắt đầu làm việc với ReactJS (2 cách).
- Học các kiến thức nền tảng của React:
 - React component
 - Props và State
 - Class Component và Functional component
 - Truyền dữ liệu giữa các components.
 - Sử dụng Style trong ứng dụng React.
 - Xử lý sự kiện trong ứng dụng React.
 - React Hooks và cách sử dụng.
 - Redux - cách quản lý state.
- Cách lấy dữ liệu từ API và hiển thị lên ứng dụng React.
- Triển khai ứng dụng React lên Internet.
- Thực hành từng bước xây dựng ứng dụng Todos.
- Bài tập: Xây dựng ứng dụng Meme Generator.

Để đảm bảo các bạn tập trung và hiểu rõ những khái niệm, kiến thức nền tảng của ReactJS, mình sẽ không sử dụng bất kỳ thư viện 3rd nào khi xây dựng ứng dụng, ngoại trừ axios và redux.

AI NÊN ĐỌC CUỐN SÁCH NÀY?

Cuốn sách này phù hợp với tất cả những ai quan tâm đến việc phát triển ứng dụng ReactJS. Với những bạn đang định hướng sự nghiệp trở thành front-end developer thì cuốn sách này dành cho bạn.

Đây là cuốn sách "**No Experience Require**", tức là không yêu cầu người có kinh nghiệm ReactJS. Tất cả sẽ được mình hướng dẫn từ con số 0.

Yêu cầu trình độ

Do ReactJS là thư viện Javascript được xây dựng để tạo các ứng dụng web, nên sẽ tốt hơn nếu bạn đã có:

- Kiến thức cơ bản về HTML và CSS.
- Javascript cơ bản (Object, Arrays, điều kiện.v.v...)
- Đã biết tới Javascript ES6 (arrow function.v.v...)

Nếu bạn vẫn còn đang bối ngỡ với Javascript, cũng không sao! Đọc xong cuốn sách này bạn hiểu Javascript hơn.

Cách học đúng cách

Cuốn sách này mình chia nhỏ nội dung thành nhiều phần, mỗi phần sẽ giới thiệu một chủ đề riêng biệt, kèm thực hành. Mục đích là để bạn có thể chủ động lịch học, không bị dồn nén quá nhiều.

Với mỗi phần lý thuyết, mình đều có ví dụ minh họa và code luôn vào dự án. Vì vậy, cách học tốt nhất vẫn là vừa học vừa thực hành. Bạn nên **tự mình viết lại từng dòng code** và kiểm tra nó trên trình duyệt. Đừng copy cả đoạn code trong sách, điều này sẽ hạn chế khả năng viết code của bạn, cũng như làm bạn nhiều khi không hiểu vì sao code bị lỗi.

"Nhớ nhé, đọc đến đâu, tự viết code đến đó, tự build và kiểm tra đoạn code đó chạy đúng không"

Ngoài ra, cuốn sách này mình biên soạn theo mô hình: phần sau được xây dựng từ phần trước. Do vậy, bạn đừng đọc lướt mà bỏ sót đoạn nào nhé.

Tất cả mã nguồn trong cuốn sách sẽ được mình up lên Github:

<https://github.com/vntalking/ebook-reactjs-that-don-gian>

Trong quá trình bạn đọc sách, nếu code của bạn không chạy hoặc chạy không đúng ý muốn, bạn có thể kiểm tra và so sánh với mã nguồn của mình trên Github. Nếu vẫn không được thì đừng ngần ngại đặt câu hỏi trên Group: [Hỏi đáp lập trình - VNTALKING](#)

Liên hệ tác giả

Nếu có bất kỳ vấn đề gì trong quá trình học, code bị lỗi hoặc không hiểu, các bạn có thể liên hệ với mình qua một trong những hình thức dưới đây:

- Website: <https://vntalking.com>
- Fanpage: <https://facebook.com/vntalking>
- Group: <https://www.facebook.com/groups/hoidaplaptrinh.vntalking>
- Email: support@vntalking.com
- Github: <https://github.com/vntalking/ebook-reactjs-that-don-gian>

GIỚI THIỆU

ReactJS (đôi lúc cũng có thể gọi là React.JS, React - cũng đều nó cả) là một thư viện Javascript giúp bạn nhanh chóng xây dựng giao diện ứng dụng. React có thể xây dựng website hoàn toàn sử dụng Javascript (để thao tác với HTML), được tối ưu hiệu năng bởi kỹ thuật **VirtualDOM** (mình sẽ giải thích chi tiết ở phần dưới cuốn sách).

React được Facebook giới thiệu vào năm 2011, họ quảng bá đây là thư viện cho phép developer tạo các dự án ứng dụng web lớn, có giao diện UI phức tạp từ một đoạn mã nguồn nhỏ và độc lập.

Một số diễn đàn, React được coi là Framework vì khả năng và behaviors (hành vi) của nó. Nhưng về mặt kỹ thuật, nó chỉ là một thư viện. Không giống như Angular hay Vue được định nghĩa là một framework. Với React, bạn sẽ cần phải kết hợp sử dụng nhiều thư viện hơn để tạo thành một giải pháp hoàn chỉnh. Điều này khiến **React chỉ tập trung vào rendering** (kết xuất - nghĩa là xuất ra một thứ gì đó cho người dùng nhìn thấy) và đảm bảo nó được đồng bộ hóa với State của ứng dụng. Đó là tất cả những gì mà React thực hiện.

Đọc mấy khái niệm này có vẻ cao siêu khó hiểu nhỉ?

Đừng hoảng nhé!

Bạn sẽ hiểu rõ hơn khi bắt tay vào thực hành xây dựng ứng dụng React ở phần tiếp theo cuốn sách - cứ bình tĩnh nhé.

Ưu điểm của React

Để tìm một thư viện/framework front-end, bạn chỉ cần vào Google gõ nhẹ một cái là ra hàng tá luôn, có thể kể những cái tên đình đám như: Angular, Vue, React, JQuery, Emberjs.v.v...

Do đó, khi phải sử dụng React, chắc hẳn bạn sẽ luôn băn khoăn: *Thư viện React này có ưu điểm gì mà mình phải dùng?*

Ở đây, mình sẽ chia thành từng mục và đưa ra lý do ngắn gọn tại sao sử dụng React là cần thiết nếu bạn đang cân nhắc đến việc xây dựng ứng dụng web hiện đại.

Ok bắt đầu nhé.

1. Component độc lập - có thể tái sử dụng

Trong React, giao diện được xây dựng từ việc kết hợp các components. Bạn có thể hiểu đơn giản component là một hàm, nó nhận giá trị bạn truyền vào và trả về một số đầu ra.

Và cũng giống như function, component có thể tái sử dụng ở bất kỳ đâu. Vì vậy, chúng ta có thể tái sử dụng lại, hợp nhất các component để tạo thành giao diện người dùng phức tạp hơn.

Trong cuốn sách này, bạn sẽ biết cách làm thế nào xây dựng giao diện ứng dụng phức tạp, nhiều tầng, nhiều lớp thông qua component.

2. React làm cho front-end javascript dễ sử dụng hơn, nhanh hơn bằng cách sử dụng Virtual DOM

Khi bạn làm việc với vanilla Javascript, bạn sẽ phải tự làm mọi nhiệm vụ khi thao tác với DOM. Nhưng với React thì khác.

Với React, bạn chỉ cần thay đổi state của UI (bạn sẽ biết khái niệm này ở phần 4 của cuốn sách), và React sẽ tự động cập nhật DOM để phù hợp với state đó. Kỹ thuật được React sử dụng đó là Virtual DOM. Điều này cho phép React chỉ cập nhật những cái cần thiết (kiểu như một phần trang web) thay vì phải tải lại toàn bộ trang.

Việc sử dụng Javascript để tạo ra mã HTML cho phép React có một cây đối tượng HTML ảo — VirtualDOM. Khi bạn tải trang web sử dụng React, một VirtualDOM được tạo ra và lưu trong bộ nhớ.

Mỗi khi state thay đổi, chúng ta sẽ cần phải có một cây đối tượng HTML mới để hiển thị lên trình duyệt. Thay vì tạo lại toàn bộ cây, React sử dụng một thuật toán thông minh để chỉ tạo lại các thành phần khác biệt giữa cây mới và cây cũ. Bởi vì cả hai cây HTML cũ và mới đều được lưu trong bộ nhớ, quá trình xử lý này diễn ra siêu nhanh.

3. Dễ dàng làm việc nhóm

Nếu bạn đang làm việc theo nhóm, thì React là thư viện tuyệt vời. Bởi vì, việc chia các task, chia màn hình ứng dụng sẽ rất đơn giản, mỗi người sẽ được chỉ định làm một thành phần. Các thành phần này độc lập với nhau, việc ghép nối các thành phần được thực hiện dễ dàng.

4. React được đảm bảo bởi Facebook

Khi bạn lựa chọn một thư viện/framework, việc đầu tiên là phải xem nó được bảo chứng bởi ai? Bởi vì một thư viện/framework không được bảo chứng bởi một tổ chức, công ty uy tín, nó có thể sẽ không được phát triển lâu dài, không được mở rộng hoặc maintain khi thư viện/framework phát sinh bugs...

Với React, bạn hoàn toàn yên tâm, vì nó được Facebook chống lưng, theo đó là một cộng đồng React tuyệt vời. Ngoài ra, tài liệu hướng dẫn chính chủ cũng rất chi tiết, đầy đủ.

Nói chung, về nguồn gốc, cộng đồng, tài liệu, bạn hoàn toàn có yên tâm gửi gắm dự án của mình cho React.

5. Nhu cầu tuyển dụng cao

Như mình đã đề cập ở trên, React là một kỹ năng web có nhu cầu tuyển dụng cao ở thời điểm hiện tại.

Hầu như nhà tuyển dụng nào khi cần tuyển một front-end developer đều yêu cầu phải biết React. Vì vậy, việc thành thạo React là một điểm sáng trong CV của bạn, cơ hội việc làm sẽ rộng mở hơn rất nhiều.

Ngoài ra, React có ít API hơn so với các thư viện/framework khác, do vậy việc học React cũng dễ hơn, đặc biệt nếu bạn đã thành thạo Javascript.

Trên đây là những ưu điểm của React mà mình tổng hợp và hệ thống lại. Giờ là lúc chúng ta cùng nhau chinh phục React thôi.

Let's go!

TỔNG QUAN REACT

Khi bạn bắt đầu tìm hiểu bất kỳ một nền tảng, thư viện hay framework nào đó, việc đầu tiên là bạn cần hiểu tư duy/triết lý của người viết ra nó. Hay nói một cách khác, bạn cần cùng hướng nhìn với tác giả của thư viện đó, khi mà cùng "hệ quy chiếu" thì việc học sẽ dễ dàng hơn.

Với React, một trong những thứ mà bạn cần hiểu đầu tiên, đó là: Virtual DOM.

Virtual DOM

Hiểu được cách thức hoạt động của DOM, bạn sẽ nhanh chóng nắm bắt được concept đằng sau Virtual DOM.

Nếu bạn là một Javascript developer, đã từng xây dựng một trang web thì hẳn bạn đã tương tác với DOM. Tuy nhiên, mình vẫn muốn nhắc lại cách thức hoạt động của nó để chúng ta cùng có một hướng nhìn.

DOM là gì?

DOM (viết tắt của Document Object Model) là một giao diện lập trình ứng dụng, cho phép Javascript hoặc các một loại script khác đọc và thao tác nội dung của document (trong trường hợp này là HTML).

Bất cứ khi nào một HTML document được tải xuống trình duyệt dưới dạng một trang web, một DOM sẽ được tạo cho trang đó.

Theo cách này, Javascript có thể thao tác với DOM như thêm, sửa, xóa nội dung... hoặc thực hiện một hành động nào đó như ẩn/hiện một view.

Vấn đề của DOM là gì?

Một số developers tin rằng, việc thao túng DOM là không hiệu quả, có thể gây chậm ứng dụng. Đặc biệt là khi bạn thường xuyên phải cập nhật DOM.