**TÀI LIỆU KHÓA HỌC “React Pro Max với Next.js”**

Tác giả: Hỏi Dân IT & Eric

Version: 8.0

Note cập nhật:

- Update source code backend (chapter 5) (v8.0)

- Bổ sung video #4.2, hướng dẫn sử dụng source code (final) của khóa học - Update link backup download source code backend (link gitlab) - Bổ sung video giải thích về cơ chế Refresh token của Next-auth

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 0: Bắt buộc xem 7** #0. Demo Kết quả đạt được 7 #1. Hướng Dẫn Download Tài liệu khóa học 11 #2. Yêu cầu của khóa học 12 #3. Về khóa học này 13 #4.1 Hướng Dẫn Sử Dụng Khóa Học Hiệu Quả 14 #4.2 Hướng Dẫn Sử Dụng Source Code của Khóa Học 15 #4.3 Về tác giả 15

**Chapter 1: Setup Environment 17** #5. Công cụ code IDE 17 #6. Browser 18 #7. Node.JS 19 #8. Quản lý code với Git 20

**Chapter 2: Ôn tập Javascript 21** #9. Cách học kiến thức Javascript hiệu quả 21 #10. Destructuring Assignment 22 #11. Spread syntax 25 #12. Conditional Operator (Toán tử điều kiện) 27 #13. Optional Chaining 28

**Chapter 3: React Basic 30** #14. Lưu ý về cách code React 30 #15. React là gì ? Phân loại dự án với React 31 #16. Setup Project React với Vite 34 #17. Cấu trúc dự án React Vite 35 #18. Think in React 36 #19. Component 37 #20. Import/Export Component 39 #21. React JSX 41 #22. Sử dụng Variables với JSX 42 #23. Bài tập design Component Input 43 #24. Props là gì 44 #25. Props với Typescript 45 #26. React Devtool 46 #27. Khái niệm Re-render 47 #28. Rendering Lists 48 #29. Key 49

**Chapter 4: React Hook 50** #30. Event Handler 50 #31. State là gì ? 51 #32. useState Hook 52 #33. Bài tập về State (part 1) 53 #34. Bài tập về State (part 2) 54

1

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

#35. Passing Function từ Cha sang Con (Parent to Child) 55 #36. Lift up State 56 #37. Vòng đời của component 57 #38. Về React Cơ Bản 58

**Chapter 5: Setup Dự Án Backend 60** #39. Giới thiệu tổng quan về dự án thực hành 60 #40. Cài đặt Postman 61 #41. Cài đặt Mongodb Compass tại local 62 #42. Setup database MongoDB Atlas 63 #43. Kích hoạt dự án backend 64 #44. Run dự án backend 65

**Chapter 6: Routing với React (Client Side) 66** #45. React Router 66 #46. Tạo Route Users 67 #47. Design table User 68 #48. Backend và API là gì 69 #49. Postman để test API 70 #50. Gọi API với React 71 #51. Mô hình stateless 72 #52. Hiển thị list User 73 #53. Sử dụng key React hiệu quả 74

**Chapter 7: Hướng dẫn sử dụng Chrome Dev Tool (Bổ Trợ) 77** #54. Thành phần của dev tool 77 #55. Kỹ thuật check api 78 #56. Nguyên tắc khi gọi API không có kết quả 79

**Chapter 8: Module Users 80** #57. Tích hợp antd 80 #58. Design giao diện Admin 81 #59. CSS với React 82 #60. Tích hợp SASS (SCSS) 83 #61. Antd Table 84 #62. Hiển thị danh sách Users 85 #63. Antd Modal 85 #64. Design Modal Add New 86 #65. API Add New User 87 #66. Bài tập Add New User 88 #67. Bài tập Design Modal Update User 89 #68. Bài tập Update User 90 #69. Design Delete User 90 #70. Bài tập Delete User 91 #71.1 Pagination 92 #71.2 Antd Pagination 94 #71.3 Update Token 94

2

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

#72. Bài tập Paginate User 95 **Chapter 9: Tối ưu hóa hiệu năng Form React 96** #73. Controlled Component vs Uncontrolled Component 96 #74. Thư viện hỗ trợ Uncontrolled Component 97 #75. Create User (Uncontrolled Component) 98 #76. Update User (Uncontrolled Component) 98 **Chapter 10: React Hiện Đại với Next.JS 99** #77. Vấn đề tồn đọng với cách viết React cũ (CSR) 99 #78. Nextjs là gì 100 #79. Nextjs làm frontend hay backend ? 101 #80. Setup dự án với Nextjs (tích hợp MUI) 102 #81. Cấu trúc dự án Next.js 103 #82.1 React children 104 #82.2 React Fragment 104 #83. Cách đọc tài liệu của Nextjs 106 #84. Chuyện Next.js ngốn RAM 107 **Chapter 11: Next Routing 108** #85. Tạo base giao diện 108 #86. Routing với nextjs 109 #87. Định nghĩa route 110 #88. Phân biệt cách code UI với React 111 #89. MUI là gì 114 #90. Cách sử dụng MUI component 115 **Chapter 12: Module HomePage 116** #91. Bài tập Design App Bar 116 #92. Navigating 117 #93. Share Layout 117 #94. Server Component 118 #95. Client Component 119 #96. Use Client 120 #97. Bài Tập Design Main Content 121 #98. Design Footer 122 #99. Nextjs Environment Variables 123 #100. Fetch music from backend 124 #101. Fetch data với Nextjs 125 #102. Fetch Data HomePage 126 #103. Fetch Wrapper 127 #104. Hoàn thiện HomePage 129 **Chapter 13: Module Wavesurfer 130** #105. Dynamic Route 130 #106. Search Params 131 #107. Giới thiệu về Wavesurfer 132 #108. Next.JS Public Folder 133

3

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

#109. Wavesurfer Basic 134 #110. React Ref 135 #111. Fetch track from Remote 136 #112. Tạo wavesurfer hook 139 #113. Sử dụng useMemo hook 140 #114. Handle Event với useCallback 141 #115. Add Bar Width 142 #116. HTML Canvas Gradients 143 #117. Add Gradients 143 #118. Add Time 144 #119. Add Hover 145 #120. Remove ID by Ref 145 #121. Add Customize Bar 146 #122. Add Comment 147 #123. Add Tooltip 148

**Chapter 14: Module Auth 149** #124. Tài liệu về Authentication với Next.js 149 #125. Giới thiệu về Next-auth 150 #126. Test Next Auth 151 #127. Debugs với Next.js và VSCode 152 #128. Config Next-Auth Routes 154 #129. Config Github Provider 155 #130. Luồng hoạt động của Next-auth ? 156 #131. Config Session Cho Client 158 #132. Config Session cho Server 160 #133. Customize Session 161 #134. Login with Credentials 166 #135. Nested Layout với Route Group 166 #136. Bài tập design Login Form 167 #137. Chữa Bài Tập Design Login 168 #138. Modify login page with Social Media 168 #139. Modify login page with Credential 169 #140. Hiển Thị Thông Báo Lỗi Khi Login 169

**Chapter 15: Module Tracks 170** #141. Bài tập hiển thị danh sách tracks (admin) 170 #142. Chữa bài tập hiển thị danh sách tracks (admin) 170 #143. Delete a track (admin) 170 #144. Ý Tưởng Design Upload Track 171 #145. Input Type File 172 #146. Tích hợp MUI Tabs 172 #147. Tích hợp Drag/Drop 173 #148. Bài tập Design Tab Information 175 #149. API Upload Single File 176

4

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

#150. Upload file with Axios 177 #151. Axios with Percents 177 #152. Upload track complete 178 #153. Add Custom Message 179 #154. Profile Page 180 #155. Bài tập render List images 180 #156. Customize track footer 181 #157. Quản lý React State Global 182 #158. Tích hợp React Context 183 #159. Play/Pause Track with Context 184 #160. Bài Tập Update view detail 186

**Chapter 16: Module Likes, Comments 187** #161. Bài tập CRUD Comments (Admin) 187 #162. Bài tập hiển thị comment trên track 187 #163. Bài tập Component Comments 188 #164. Bài tập add a comment 189 #165. Bài tập add likes 190 #166. Add count view 191

**Chapter 17: Module SEO (Search Engine Optimization) 192** #167. Điều cần biết về SEO 192 #168. Công cụ check SEO 193 #169. Metadata 194 #170. SEO 1: Tối ưu hóa project 195 #171. SEO 2: Đặt tên Title/Description (Static) cho web page 196 #172. SEO 3: Đặt tên Title/Description (Dynamic) cho web page 197 #173.1 Cách hosting ảnh FREE (Bonus) 198 #173.2 SEO 4: Share link website trên Facebook/Twitter 199 #174. SEO 5: Thêm logo cho website 201 #175. SEO 6: Url với slug 202 #176. SEO 7: Robot.txt, sitemap và manifest.json 203 #177. SEO 8: JSON-LD 204 #178. SEO 9: Next.js Image 205

**Chapter 18: Advance - Tối ưu dự án Next.JS 209** #179. Add Progress Bar 209 #180. Handle Error 209 #181. Middleware 210 #182. Add Loading 211 #183. Nextjs Build Script 212 #184. Build với Docker 214 #185. Phân tích kết quả Next.js build 216 #186. SSR và Static Site Generation 217 #187. SSG và generateStaticParams 218 #188. NextJS Caching 219

5

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

#189.1 Revalidating data (Time based revalidation) 221 #189.2 Revalidating data (On-demand revalidation) 223 **Chapter 19: Module Playlists 225** #190. Bài tập Playlist Page 225 #191. Bài tập Likes page 227 #192. Bài tập Chức năng search 228 #193. Login with Google 229 #194.1 Fix bug dự án 230 #194.2. Upgrade Next.js Version (và các thư viện khác) 231 #194.3 Next-Auth và Refresh Token 232 **Chapter 20: What's next ? 233** #195. Why NextJS ? 233 #196. Nhận xét về Level Fresher 234 #197. Nhận xét về khóa học này 235 #198. What’s next - Học gì tiếp theo ? 236 **Chapter 21: Update Next.js 14 với Antd (NEW) 237** #199. Chuyện công nghệ update 237 #200. Một vài lưu ý về Next.js Update 239 #201. Next.js 14 có gì hot ? 241 #202. Setup Next.js 14 Only 242 #203. Setup Antd 243 #204. Bài tập Design Login Form với Antd 243 #205. Khái niệm Flick giao diện 244 #206. Setup Antd với Render trên Server 245 **Chapter 22: Server Actions (NEW) 246** #207. Tổng quan về Server Actions 246 #208. Handle Form với Server 247 #209. Loading State và useFormStatus Hook 248 #210. Handling Response Data và useFormState Hook 249 #211. Các công cụ sử dụng với Server Actions 250 #212. Call Server Actions from Functions ? 251 **Chapter 23: Thực Hành Mutate Data với Server Actions (NEW) 252** #213. Design Giao Diện CRUD 252 #214. Bài Tập Create User Actions 253 #215. Bài Tập Update User Actions 253 #216. Bài Tập Delete User Actions 253 #217. Update Dự Án SoundCloud Sử Dụng Next.js 14 254 #218. So sánh Server Actions và Route Handler 255 #219. Tổng kết về Server Actions 257 #220. Tự Học Fullstack Next.js/Nest.js (Extra) 257 **Lời Kết 258**

6

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 0: Bắt buộc xem**

*Giới thiệu và demo kết quả đạt được sau khi kết thúc khóa học này.*

**#0. Demo Kết quả đạt được**

Link video demo: https://youtu.be/dsha7HgqMhI

Về frontend, mình đã cover:

- Bootstrap (React Ultimate)

- Ant Design (React Test Fresher)

- MUI (React Pro Max & Next.js)

1. Demo thành quả đạt được

Dự án clone soundcloud (website nghe nhạc)

**Tính năng chính:**

+ Đăng nhập : hỗ trợ đăng nhập với Google/Github và tài khoản local của backend. Có khả năng hỗ trợ đăng nhập với các nền tảng được liệt kê tại đây: https://next auth.js.org/providers/

+ Nghe nhạc với:

- audio track ở footer

- wavetrack khi xem chi tiết

+ Comment theo thời gian của wavetrack

+ Tạo playlist/Like tracks

+ Với giao diện admin: CRUD user/tracks/comment...

7

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

2. Công nghệ sử dụng

**Frontend:**

- Client: Nextjs, MUI (**typescript**). Login with Next-auth (hỗ trợ đăng nhập Google/Github và Credential Provider (backend))

- Admin: Vite, Antd (typescript)

**Backend:** Nestjs (được cung cấp sẵn)

**Về kiến thức React** (typescript với Vite), ngoài kiến thức cơ bản, bổ sung thêm: - React Children, React Fragment, React Context. Suspense/Lazy loading - Các hook: useRef, useCallback, useMemo

**Về kiến thức Nextjs:**

- Nextjs 13 với App Router (typescript)

- Nắm vững các kiến thức thay đổi của Nextjs:

+ Routing (dynamic routes, route groups)

+ Data Fetching (Server component vs Client component (use client))

+ Authentication with Next-auth (sử dụng Session/JWT/Refresh Token) + SEO (mức basic, test với lighthouse built-in extension)

+ Rendering:

client side rendering (CSR), server side rendering (SSR),

static site generation (SSG), incremental Static Regeneration (ISR)

+ Chế độ build:

- build với docker

- demo revalidate Tag after build :v

- Sử dụng MUI để base component

8

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

3. Đối tượng theo học

- Bạn "bắt buộc" cần biết HTML, CSS và cú pháp javascript. Học viên bắt đầu từ số 0 theo học không phù hợp.

- Bạn không cần phải biết code Backend (vì backend được cung cấp sẵn).

Nếu được, bạn có thể xem qua 2 series: (hoàn toàn miễn phí trên Youtube Hỏi Dân IT) - React cơ bản

- Typescript

-> 2 series trên sẽ giúp bạn có nền tảng vững chắc để theo học.

Phần còn lại, khóa học sẽ giúp bạn:

- Học kiến thức React.js (Typescript)

- Học Nextjs (Typescript)

**Nếu bạn là người đã có tư duy lập trình (đã đi làm), cần 1 khóa học để end-game với React**

=> đây chính là khóa học bạn cần.

Nếu chưa chắc chắn, cần tư vấn & hỗ trợ giải đáp thắc mắc, inbox Facebook Hỏi Dân IT: https://www.facebook.com/askITwithERIC

4. Tài liệu khóa học

- Cung cấp qua pdf, được dùng để 'take note' kiến thức quan trọng, link source code, link ví dụ...

- Được đánh số version, link download trực tiếp trong khóa học

- Lưu ý về cách sử dụng bookmark khi xem online

5. Source code Frontend

**Source code Frontend cả khóa học "được" cung cấp full, mình sẽ hướng dẫn tại các video tiếp theo**

9

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

6. Source code Backend

- **Source code backend bạn sẽ được cung cấp sẵn, dưới dạng mã hóa**. Chỉ chạy và không sửa đổi.

Lý do: Đây là khóa học frontend, nên sẽ không học backend.

Bạn muốn biết source code này viết như nào, vui lòng học lộ trình backend

7. Chuyện deploy

- Khóa học này chạy localhost, vì hosting free ngoài kia rất "lởm". hàng free mà.

- Do khóa học này có backend và việc thao tác với file => hosting free không có khả năng lưu trữ file

- Deploy chuyên nghiệp cần biết cả frontend và backend

- Trong cv, bạn ghi link github là đủ. nếu cần thiết, thì quay video demo, thay vì deploy free

- Muốn deploy thực tế, bạn cần học kỹ năng backend, sau đấy deploy với server trả phí. Link khóa học deploy: https://hoidanit.com.vn/khoa-hoc/ultimate-guide-to-deploy-react nodejs-640bee82f7099c369b3bc6a4.html

8. Chuyện bản quyền/học chung

Khóa học được đánh bản quyền theo thỏa thuận giữa bạn và tác giả Hỏi Dân IT. Mọi hành vi "share/mua chung" đều được coi là vi phạm => đồng nghĩa với việc bạn đang tước bỏ đi quyền lợi của học viên

Nếu bạn có nhu cầu học chung (mua chung), cứ inbox mình. Chúng ta sẽ tìm được tiêng nói chung (mình có hỗ trợ khi mua với số lượng lớn)

9. Hình thức mua

- Inbox trực tiếp fanpage Hỏi Dân IT: https://www.facebook.com/askITwithERIC

- Luôn có chính sách ưu đãi cho học sinh/sinh viên/người chưa đi làm và các học viên cũ

10

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#1. Hướng Dẫn Download Tài liệu khóa học**

Link download tài liệu khóa học được cập nhật theo thời gian, các bạn lưu ý về version của tài liệu để biết được tài liệu là cũ hay mới nhé.

(So sánh tên version của tài liệu khóa học của bạn, để biết được nó là cũ hay mới) Link tài liệu khóa học tải tại đây

11

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#2. Yêu cầu của khóa học**

1. Biết HTML, CSS và cú pháp Javascript (tự học)

Về HTML, CSS => tự học

Về Javascript, tham khảo: (chỉ cần học để hiểu cú pháp khai báo của javascript) https://www.w3schools.com/js/

Link khóa học Javascript miễn phí:

https://www.youtube.com/playlist?list=PLncHg6Kn2JT5dfQqpVtfNYvv3EBVHHVKo

2. Biết cú pháp typescript:

Link khóa học Typescript miễn phí:

https://www.youtube.com/playlist?list=PLncHg6Kn2JT5emvXmG6kgeGkrQjRqxsb4

3. Biết sử dụng Git để quản lý mã nguồn

Link khóa học Git miễn phí:

https://www.youtube.com/playlist?list=PLncHg6Kn2JT6nWS9MRjSnt6Z-9Rj0pAlo

12

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#3. Về khóa học này**

Mục tiêu:

**- Chỉ 1 khóa học duy nhất, học React & Next.JS**

- Giúp định hướng và rút ngắn thời gian học kiến thức, phục vụ mục đích đi thực tập & đi làm

Khóa học này có "bị trùng nội dung" với lộ trình FE đã có của mình không ? **Câu trả lời là không.**

**Điểm khác biệt của khóa học này, là học thêm Next.js**

Tuy nhiên, để học Nextjs, bạn cần "biết React trước".

=> các khóa cũ, là học "mình React". điều này không có trong khóa Next.js (từ kiến thức cơ bản tới nâng cao, cho tới việc dùng với Typescript)

=> học xong khóa next.js này, recommend (gợi ý) học thêm các khóa FE còn lại để hiểu sâu hơn về React :v

13

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#4.1 Hướng Dẫn Sử Dụng Khóa Học Hiệu Quả**

Source code frontend phần React Vite (hết video #76):

https://drive.google.com/file/d/1RAWwfB1bqQR\_mdmExChIIhPByiETpg6j/view?usp=sh aring

Lưu ý: với source code trên, các bạn cần tự setup và xem giúp mình chapter 5 để biết cách chạy backend cho dự án

Hoặc: có thể sử dụng source code trên để đỡ phải code, trong khi vẫn xem đầy đủ video của khóa học

Mục đích cung cấp source code, là dành cho các bạn “đã biết code React với Typescript” (ví dụ như là đã học qua lộ trình frontend của mình rồi chẳng hạn).

Tất cả trường hợp khác, mình KHÔNG KHUYẾN KHÍCH sử dụng, thay vào đấy là học theo trình tự của khóa học

=> kiến thức được liền mạch và tránh bị hổng kiến thức

14

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#4.2 Hướng Dẫn Sử Dụng Source Code của Khóa Học**

Đây là khóa học thực hành, mình “KHÔNG khuyến khích” sử dụng source code. Vì nếu các bạn “không tự code”, kiến thức không bao giờ (never) là của các bạn. Đi làm, ai code hộ bạn ?

Link source code của khóa học (final):

https://gitlab.com/public-starter-projects1/01-nextjs-basic-soundcloud/00-final project/00-nextjs-mui-soundcloud-final

Câu lệnh để lấy code tại từng video:

**git checkout mã-hash-của-commit**

**#4.3 Về tác giả**

Mọi thông tin về Tác giả Hỏi Dân IT, các bạn có thể tìm kiếm tại đây: Website chính thức: https://hoidanit.vn/

Youtube “Hỏi Dân IT” : https://www.youtube.com/@hoidanit Tiktok “Hỏi Dân IT” : https://www.tiktok.com/@hoidanit

Fanpage “Hỏi Dân IT” : https://www.facebook.com/askITwithERIC/ Udemy Hỏi Dân IT: https://www.udemy.com/user/eric-7039/

15

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Về chuyện leak khóa học và mua lậu**

Mình biết rất nhiều bạn khi học khóa học này của mình, là mua lậu qua bên thứ 3. chuyện này là hoàn toàn bình thường, vì thương hiệu “Hỏi Dân IT” đang ngày càng khẳng định được vị thế của mình.

Nhiều bạn hỏi mình, sao mình không ‘chặn việc mua lậu’. nói thật, nếu mình làm, là làm được đấy, cơ mà nó sẽ gây ra sự bất tiện cho học viên chân chính (con sâu làm rầu nồi canh). Với lại, ngay cả hệ điều hành windows, còn bị crack nữa là @@

Mình cũng có 1 bài post facebook về chuyện này:

https://www.facebook.com/askitwitheric/posts/pfbid02gyasktd3semgxat6nevnvwh4c 8epzu3i7kpzhr7s7gmmfcvucyz96eb8avnvgwnhl

Với các bạn học viên chân chính, mình tin rằng, những cái các bạn nhận được từ mình khi đã chấp nhận đầu tư, nó sẽ hoàn toàn xứng đáng. vì đơn giản, với cá nhân mình, khách hàng là thượng đế.

VỚI CÁC BẠN MUA LẬU, MÌNH CHỈ MUỐN CHIA SẺ THẾ NÀY:

**1. TRÊN ĐỜI NÀY, CHẲNG CÓ GÌ CHẤT LƯỢNG MÀ MIỄN PHÍ CẢ.** VIỆC BẠN MUA LẬU QUA BÊN THỨ 3, LÀ GIÚP BỌN CHÚNG LÀM GIÀU VÀ GÂY THIỆT HẠI CHO TÁC GIẢ.

NẾU NHÌN VỀ TƯƠNG LAI => CÀNG NGÀY CÀNG ÍT TÁC GIẢ LÀM KHÓA HỌC => NGƯỜI BỊ HẠI CUỐI CÙNG VẪN LÀ HỌC VIÊN

**2. HÃY HỌC THÓI QUEN TRÂN TRỌNG GIÁ TRỊ LAO ĐỘNG**

NÓ LÀ THÓI QUEN, CŨNG NHƯ SẼ LÀ MỘT PHẦN TÍNH CÁCH CỦA BẠN. ĐỪNG VÌ NGHÈO QUÁ MÀ LÀM MẤT ĐI TÍNH CÁCH CỦA BẢN THÂN. NẾU KHÓ KHĂN, CỨ INBOX MÌNH, MÌNH HỖ TRỢ. VIỆC GÌ PHẢI LÀM VẬY =))

3. **MÌNH ĐÃ TỪNG LÀ SINH VIÊN GIỐNG BẠN**, MÌNH HIỂU TẠI SAO CÁC BẠN LÀM VẬY. **HÃY BIẾT CHO ĐI. SỐNG ÍCH KỶ, THÌ THEO LUẬT NHÂN QUẢ ĐẤY, CHẲNG CÓ GÌ LÀ NGẪU NHIÊN CẢ**

**4. NẾU BẠN THẤY KHÓA HỌC HAY, HÃY BIẾT DONATE ĐỂ ỦNG HỘ TÁC GIẢ. LINK DONATE: https://hoidanit.com.vn/donate**

**Hành động nhỏ nhưng mang ý nghĩa lớn. Hãy vì 1 cộng đồng IT Việt Nam phát triển. Nếu làm như các bạn, có lẽ chúng ta đã không có Iphone, không có Apple như ngày nay rồi @@**

16

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 1: Setup Environment**

*Cài đặt các công cụ cần thiết cho khóa học*

**#5. Công cụ code IDE**

Cài đặt Visual Studio Code (VSCode):

Link download: https://code.visualstudio.com/download

- Setup VSCode: auto format on save

- Setup extension: (mục đích setup extension là "hỗ trợ" việc code, không phải là "phụ thuộc" vào extension)

**=> chỉ setup extension "cần thiết", và hạn chế tối đa sự phụ thuộc vào extension**

**+ auto complete tag, auto close tag, auto rename tag**

=> hỗ trợ việc code html nhanh hơn

**+ code spell checker**

=> hỗ trợ check tên biến/hàm có bị sai chính tả hay không ? (lưu ý là check tiếng anh, ko phải là check tiếng việt)

- Lưu ý về extension eslint, pretty (nếu cài: disabled)

Nếu bạn "từng học ở các nguồn khác", họ hướng dẫn cài đặt eslint, rồi pretty để "code cho đẹp hơn", thì vui lòng disabled những extension này, để tránh những "báo lỗi" không cần thiết.

Eslint, hay pretty, là cách bạn đặt ra quy tắc code, nhưng mà, mỗi cá nhân, mỗi công ty, có 1 quy định khác nhau

=> nếu bạn muốn setup eslint, hay pretty, thì setup trực tiếp trong project, không setup qua extension bạn nhé

**Ở đây, chúng ta disabled, mục đích là dùng "những format" được quy định sẵn khi dùng VScode (không cần cài đặt extension).**

17

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#6. Browser**

Về browser (trình duyệt web), bạn vui lòng tuân theo quy định bên dưới nhé :

1. Sử dụng Google Chrome (bắt buộc):

Download: https://www.google.com/chrome/

**Lưu ý không dùng Cốc Cốc, Firefox hay Edge..., vì:**

+ trong khóa học, mình sử dụng Google Chrome. Dùng giống nhau, sẽ có giao diện code giống nhau => hạn chế lỗi "không cần thiết"

+ **Google Chrome phổ biến nhất**, đi làm, chúng ta test cũng ưu tiên Google Chrome đầu tiên (cứ phổ biến mà xài)

Fact: Nếu bạn để ý, bạn sẽ thấy Firefox, hay Edge, đều bắt chước tính năng dành cho developer của Google Chrome :v

2. Chuyển ngôn ngữ Tiếng Anh (bắt buộc):

**Bạn nào dùng ngôn ngữ tiếng việt => vui lòng chuyển qua tiếng anh, vì:** + Giao diện ngôn ngữ là khác nhau. Các "thuật ngữ" ngành IT, dịch sang tiếng việt, nó không sát nghĩa (không đúng)

+ Đi làm thực tế, không ai dùng tiếng việt đâu bạn. sự thật đấy (nếu bạn không muốn được gọi là đứa nhà quê, hai lúa :v)

+ dùng tiếng anh, nó có lợi ích về lâu dài (giúp bạn đi nhanh hơn, vì tài liệu = tiếng anh luôn có sắn)

+ Nếu bạn chưa biết gì về tiếng anh => dùng google translate.

Ở đây, chưa biết gì, thì code theo

code nhiều thành quen thôi. giống bạn chơi game ấy, toàn từ ngữ tiếng anh, sao bạn chơi được ? (học code nó tương tự bạn nhé)

3. Hướng dẫn chuyển đổi ngôn ngữ của Google Chrome

//todo (trong video đã hướng dẫn)

18

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#7. Node.JS**

1. Node.JS là gì ?

**Bạn học React, tại sao lại cần Node.JS ?**

**Node.js là platform** giúp bạn có thể thực thi ngôn ngữ Javascript trên máy tính (server)

Điều này tương tự với:

Bạn muốn học **Microsoft Words** => bạn cần cài đặt hệ điều hành **Windows** (Bạn học **React** => cần cài đặt **Node.JS)**

như vậy, platform là "môi trường" giúp bạn chạy code.

chúng ta không học "node.js", mà học "react'.

Node.js là điều kiện "bắt buộc" để chúng ta chạy code các bạn nhé.

2. Cài đặt Node.JS

**Lưu ý: không cài đặt version Node.js mới nhất, vì rất nhiều thư viện chưa kịp hỗ trợ Node.js latest. Các bạn vui lòng cài version 18.17.0 để thực hành.**

Việc cài chính xác 1 version của Node.js, sẽ giúp code của mình và bạn giống nhau 100% (hạn chế tối đa lỗi không cần thiết)

Điều này tương tự với việc, bạn "chơi 1 game trên windows 7 rất ok", nhưng điều này "không chắc chắn" khi bạn dùng windows 11, right ?

- Cài node v18.17.0: https://nodejs.org/download/release/v18.17.0/

- Muốn dùng nhiều version (tránh ảnh hưởng tới dự án cũ => nvm): https://www.youtube.com/watch?v=ccjKHLyo4lM&list=PLncHg6Kn2JT6E38Z3kit9Hnif 1xC\_9VqI&index=40

3. Kiểm tra cài đặt Node.js

dùng câu lệnh: **node -v**

khi cài đặt node.js => đồng nghĩa với việc bạn cài đặt npm (node package manager) => công cụ giúp cài đặt thư viện code js

=> kiểm tra bằng cách: **npm -v**

19

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#8. Quản lý code với Git**

Nếu bạn chưa biết gì về Git, vui lòng xem nhanh tại đây:

https://www.youtube.com/playlist?list=PLncHg6Kn2JT6nWS9MRjSnt6Z-9Rj0pAlo

Mục đích sử dụng Git:

**- "Kéo code" dự án thực hành**

**- Lưu trữ "backup" cho code. Tránh tình huống máy tính bị hư => mất hết code - Git là "công cụ bắt buộc phải biết" khi đi làm thực tế (đi thực tập/fresher)**

20

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 2: Ôn tập Javascript**

*Ôn tập các kiến thức nền tảng của javascript hay được sử dụng với React. Mục tiêu của chương, là khi nào bạn cần, sẽ giải đáp ngay lập tức thắc mắc của bạn.*

**#9. Cách học kiến thức Javascript hiệu quả**

Mục tiêu của chương, là giúp bạn "nhìn mặt chữ & nhận dạng", không phải là giúp bạn thực hành ngay lập tức

Chương này không phải học lại javascript từ đầu, mà là định hướng các cú pháp "hay dùng" (thậm chí là đi làm) bạn hay gặp phải.

- Sau này học react, nếu bạn quên, có thể xem lại để hiểu hơn. Vì suy cho cùng, React cũng là javascript mà thôi

- Không ai có thể hiểu & vận dụng tất cả mọi thứ ngay từ đầu.

**Tất cả đều cần "thực hành"**

Nếu bạn quên, thì bỏ qua. Miễn sao nhớ được tên "của kiến thức" là được. sau đấy xem lại => thực hành

Chỉ có thực hành "đủ nhiều", mới thấy nó dễ.

21

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#10. Destructuring Assignment**

Tài liệu: https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Operators/Destructuring\_assignment

**Destructuring Assignment** là cách chúng ta giản lược cú pháp viết code (giúp viết code ngắn & thuận tiện hơn) khi thao tác với **array** và **object**

1. Object Destructuring

Tài liệu: https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Operators/Destructuring\_assignment#object\_dest ructuring

Ví dụ:

const user = {

name: "Eric",

age: 25,

address: "Ha Noi"

}

before:

với cách viết truyền thống, để lấy ra 3 biến name, age và address, cần sử dụng:

const name = user.name;

const age = user.age;

const address = user.address;

=> càng nhiều biến (variable), số lượng dòng code sẽ càng nhiều.

after:

const { name, age, address } = user;

=> sử dụng duy nhất đúng 1 dòng. Chúng ta dùng const { } vì user là một object. keyword const giúp khởi tạo kiểu giá trị (type) cho các biến bên trong { }

Với object destructuring => sử dụng dấu { }

22

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

2. Array Destructuring

Tài liệu: https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Operators/Destructuring\_assignment#array\_destr ucturing

Ví dụ:

const level = ["internship", "fresher", "junior", "middle", "senior" ]

before:

const internship = level[0]

const senior = level[4]

after:

const [internship, fresher, junior, middle, senior] = level;

=> tương tự như object destructuring, điểm khác biệt duy nhất là dùng [ ] thay vì { }

//Part 2

3. Ví dụ với React

**Lưu ý: không cần hiểu props, useState là gì (vì sau này sẽ hiểu), điều quan trọng là bạn nhận ra ý nghĩa của cách viết code.**

Ví dụ 1:

function User (props) {

const {name, age, address} = props;

// nhìn vào cách viết này, bạn sẽ hiểu props là 1 biến object, và chúng ta đang sử dụng object destructuring

//thay vì viết:

const name = props.name;

const age = props.age;

....

}

23

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

Ví dụ 2:

function Level ( ){

const [level, setLevel] = useState(...)

//đây là array destructuring. Không cần biết useState() là cái gì, cơ mà chúng ta biết nó là 1 array

và ở đây:

level === useState( )[0] => phần tử đầu tiên của array

setLevel === useState( ) [1] => phần tử thứ 2 của array

}

24

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#11. Spread syntax**

Tài liệu:

https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Operators/Spread\_syntax

Hiểu 1 cách đơn giản, spread syntax (toán tử mở rộng) giúp bạn có thể "copy" đối tượng (array, object).

1. Sử dụng với array

const arr1 = [1, 2, 3]

muốn thêm phần tử 4 (vị trí thứ 4), và gán nó vào ar2 thì làm sao ?

https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Global\_Objects/Array/push

before:

const arr1 = [1, 2, 3];

**arr1.push(4)**

const arr2 = arr1;

lưu ý: không viết là const arr2 = arr1.push(4)

(vì hàm push trả về length của arr => arr2 = length new array)

after:

const arr2 = [ ...arr1, 4 ] => thêm phần tử vào cuối mảng

const arr2 = [ 4, ...arr1 ] => thêm phần tử vào đầu mảng (arr.unshift)

2. Sử dụng với object

const obj1 = { foo: "bar", x: 42 };

const obj2 = { foo: "baz", y: 13 };

const clonedObj = { ...obj1 };

// { foo: "bar", x: 42 }

const mergedObj = { ...obj1, ...obj2 };

// { foo: "baz", x: 42, y: 13 }

25

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

Lưu ý: không nên làm dụng spread syntax, vì đây là shallow copy, không phải là deep copy

Tham khảo: https://hoidanit.com.vn/khoa-hoc/react-nang-cao-lieu-ban-da-hieu react.html?id=640be92bf7099c369b3bc685

Ví dụ 1: (ok)

const first\_person = {

name: "Jack",

age: 24,

}

const second\_person = { ...first\_person };

second\_person.age = 25;

console.log(first\_person.age); // output: 24

console.log(second\_person.age); // output: 25

Ví dụ 2: (not ok - sử dụng nested object)

const first\_person = {

name: "Jack",

age: 24,

address: {

apartment: "A",

city: "London"

}

};

const second\_person = { ...first\_person };

second\_person.age = 25;

second\_person.address.apartment = "N";

console.log(first\_person.address.apartment); // output: N

console.log(second\_person.address.apartment); // output: N

26

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#12. Conditional Operator (Toán tử điều kiện)**

Tài liệu: https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Operators/Conditional\_operator Toán tử điều kiện tương tự như câu điều kiện if, gồm 3 phần:

condition ? exprIfTrue : exprIfFalse

Nếu "điều kiện" là đúng, <exprIfTrue> sẽ được thực thi. ngược lại, <exprIfFalse> sẽ được thực thi.

Việc sử dụng toán tử điều kiện, again, giúp code của bạn ngắn hơn thôi :v Ví dụ:

before:

const a = 10;

let b = null;

if(a > 5){

b = true;

} else {

b= false;

}

after:

a = 10;

const b = a > 5 ? true : false;

const age = 26;

const beverage = age >= 21 ? "Beer" : "Juice";

console.log(beverage); // "Beer"

27

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#13. Optional Chaining**

https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Operators/Optional\_chaining

Optional Chaining là các truy cập thuộc tính của object (hoặc function) mà không muốn ném ra lỗi (nếu có)

Lưu ý: Không nên lạm dụng cách làm này (chỉ dùng, khi bạn biết bạn đang làm gì) 1. Sử dụng với variable

const adventurer = {

name: 'Alice',

cat: {

name: 'Dinah'

}

};

before:

const dogName = adventurer.dog.name;

console.log(dogName); //error

=> throw an error

after

const dogName = adventurer?.dog?.name;

console.log(dogName); //undefined

2. Sử dụng với function

Ví dụ:

const a = undefined;

a.forEach(item => ...)

a?.forEach(...)

28

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

Part 2:

3. Gán giá trị mặc định

https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Operators/Nullish\_coalescing

29

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 3: React Basic**

*Học các kiến thức cơ bản và cốt lõi nhất của React thông qua các ví dụ thực tế.*

**#14. Lưu ý về cách code React**

1. Phong cách code (free styles)

- Cách đặt tên biến

- Cách khai báo function

- Cách dùng dấu chấm, dấu phẩy (dùng dấu chấm phẩy cuối mỗi dòng code, tương tự như code java => tránh lỗi không cần thiết)

- không dùng extension (hạn chế tối đa nhất có thể)

=> mình không phải là "master code", tuy nhiên, mình đã trải qua cơ số công ty, và mỗi công ty đều có "**convention**" khi code.

**Vì vậy, nếu bạn là beginners, tốt nhất hãy tập code theo mình. Đi làm, người ta chỉ yêu cầu hơn, chứ không kém bạn nhé.**

2. Tài liệu học React

- Về trang tài liệu chính thống:

Từ tháng 4/2023: https://react.dev/

Trước tháng 4/2023 : **reactjs.org** or **legacy.reactjs.org**

=> tài liệu trên website này không còn được update nữa.

tuy nhiên, các **core function (kiến thức React) luôn không đổi.**

Phần update (khác biệt giữa website "mới" và "cũ"), là cách trình bày tài liệu.

cũng như các tính năng mới của react (ví dụ như react action, react server...) sẽ không có trên trang tài liệu cũ.

30

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#15. React là gì ? Phân loại dự án với React**

Part 1: React là gì

**React là “library” (thư viện) viết bằng ngôn ngữ Javascript,** giúp việc code UI (giao diện web) trở nên dễ dàng và tiện lợi hơn

**Library**: tập hợp các khối code (block code/ functions)

**Framework**: thiên về cách tổ chức code (architecture/kiến trúc), đồng thời đã làm sẵn rất nhiều tính năng.

**Platform**: Môi trường chạy code

React là “library” javascript => kiến thức về javascript bạn cần biết trước. Bạn code React, “bản chất” là bạn đang code ngôn ngữ javascript

React được Facebook chống lưng và phát triển. Amazing

1. So sánh React, Angular và Vue

https://npmtrends.com/@angular/core-vs-react-vs-vue

Lựa chọn React, vì nó phổ biến nhất

=> nguồn tài liệu phong phú, cộng đồng support nhiều

2. Chuyện 1m2 20 ông React :v

Câu chuyện này tương tự, doanh nghiệp luôn than thở, “lúc nào cũng cần/thiếu nhân sự”, trong khi người tìm việc không kiếm được việc làm.

Ở đây, bạn nên hiểu, “doanh nghiệp cần”, là cần người “làm được việc”

1m2 20 ông React, cơ mà ông nào cũng giống ông nào, chỉ biết basic (ví dụ viết Hello world), thất nghiệp là chuyện “quá bình thường”

31

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

Part 2:

1. Lịch sử ra đời

Hiện nay, react có 2 loại chính là: **React ở client** và **React ở Server**

+ React ở Client: create-react-app, vite

+ React ở Server: next.js, remix, gatsby

Why ?

+ **Trước 2012**, khi "các library/framework" chuyên làm FE chưa ra đời (React, Angular, Vue), website thông thường là full-stack.

điều này có nghĩa là, bạn code nguyên 1 website, sử dụng đúng 1 ngôn ngữ (java, php, C#...)

**Nhược điểm của cách làm này, là code FE "rất khổ", vì đa phần là code thuần html với "view engine"**

=> tính mở rộng và tái sử dụng code không cao (UI), trải nghiệm của người dùng "không sướng" (UX)

ví dụ về UX: chat real-time

Với React: (ra đời chính thức vào năm 2013)

Mục tiêu của React là giúp code FE "sướng hơn và nhàn hơn" cách code HTML truyền thống.

+ **Từ 2012 tới 2022**: sự phát triển của React (phía client)

+ **Từ 2022 -> nay (trending):** React phía server

=> giống hệt như thời trước 2012.

Điểm khác biệt, là "bạn không cần reload trang web để xem sự thay đổi :v)

32

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

2. React ở Client

- Không đặt nặng về SEO

- Quan trọng việc "thay đổi dữ liệu thường xuyên" ở client

- Dữ liệu sẽ được 100% client xử lý => loading khi vào trang

ví dụ: https://iboard.ssi.com.vn/

3.React ở Server

- Quan tâm tới SEO

- Dữ liệu ít thay đổi

- Dữ liệu sẽ được server xử lý => không bị loading khi vào trang

ví dụ: https://hoidanit.com.vn

4. Chia sẻ câu chuyện website Hỏi Dân IT

- code html thuần (2h)

- code với react vite (client) (2 tuần)

- code với next.js (server) (1 tháng) => thời gian lâu hơn, vì có sự thay đổi về thư viện (antd => mui)

33

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#16. Setup Project React với Vite**

Chúng ta sẽ học cả React ở client và React ở server

=> đi công ty nào dùng React cũng sống được

Bắt đầu từ React client, rồi tới React server (chính là lịch sử phát triển của React) => hiểu tại sao công nghệ nó lại thay đổi.

1. Setup dự án

Tài liệu: https://vitejs.dev/guide/

Lưu ý: chỉ làm theo link tài liệu, khi bạn "đã có kiến thức về react", hiểu rõ bạn đang làm gì.

**Link source code:** https://gitlab.com/public-starter-projects1/01-nextjs-basic soundcloud/react-vite-starter

Lưu ý: đã cấu hình eslint để check lỗi (react overlay):

https://github.com/vitejs/vite/issues/2076

2. Run Hello World

+ cài đặt thư viện cần thiết: npm i (hoặc gõ đầy đủ: npm install)

Lưu ý về message thông báo:

https://stackoverflow.com/questions/69885467/i-have-npm-vulnerability-issue added 149 packages, and audited 150 packages in 33s

37 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

+ chạy dự án với câu lệnh: npm run dev

Mặc định, project vite chạy tại địa chỉ: http://localhost:5173

34

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#17. Cấu trúc dự án React Vite**

1.Folder:

- node\_modules: đây là nơi "code của thư viện cài đặt". Tất cả thư viện được sử dụng trong dự án sẽ được lưu tại đây

- public: như tên gọi (công khai) => đây là nơi bạn lưu trữ file và public ra ngoài (ví dụ css/js/images...)

- src : đây là nơi lưu code dự án (không public ra ngoài)

2. Files

vite.config.ts => file cấu hình dự án với Vite

tsconfig.json và tsconfig.node.json => cấu hình typescript cho dự án README.md => guide

package.json => đây là file cấu hình tất cả câu lệnh chạy dự án, cũng như thư viện cài đặt

Nguyên tắc: cài thêm 1 thư viện mới => update file package.json

package-lock.json: file lưu chi tiết thông tin thư viện cài đặt (version cài đặt) .gitignore : quy định các file không đẩy lên git

.eslintrc.cjs: cấu hình eslint (phục vụ mục đích check code)

index.html : đây là file project sẽ chạy

<div id="root"></div>

=> nội dung code React sẽ được dịch và chèn vào div root ở trên

35

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#18. Think in React**

Tài liệu:

https://react.dev/learn/thinking-in-react

- Khái niệm component, why ?

- Chia nhanh layout, giống chơi lego

- Giúp tái sử dụng code

- Component >< HTML ở chỗ, có thể code được javascript bên trong html

36

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#19. Component**

Tài liệu: https://react.dev/learn/your-first-component

Lưu ý: viết hết trong file main app (chưa sử dụng cú pháp import/export)

1. Định nghĩa component

- Component, suy cho cùng, là javascript function

Tên file: .js , .jsx (dùng với javascript)

.ts, .tsx (dùng với typescript)

Gồm 2 thành phần:

+ logic javascript

+ template (jsx)

=> Component >< HTML ở chỗ, có thể code được javascript bên trong html

2. Nguyên tắc Khi báo function :

- Tên function cần viết hoa chữ cái đầu tiên

- cần có hàm return (template/html) để biến nó thành react component

MyFunction(){

return(

<div> </div>

)

}

37

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

Part 2:

Lưu ý về cách viết code:

1. không dùng cách viết inline

return <img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />; 2. luôn sử dụng dấu ( ) với hàm return

nên nhớ, component là function => return () có nghĩa là function này "chắc chắn/always) trả về giá trị bên trong ()

before:

return <img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />;

after:

return (<img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />);

38

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#20. Import/Export Component**

Tài liệu:

https://react.dev/learn/importing-and-exporting-components

Về cú pháp export:

https://developer.mozilla.org/en-US/docs/web/javascript/reference/statements/export

**Import**/**Export** là cách "reuse - tái sử dụng" functions/variables giữa các "file" của javascript

Ví dụ:

// say.js

function sayHi(user) {

alert(`Hello, ${user}!`);

}

function sayBye(user) {

alert(`Bye, ${user}!`);

}

**export** {sayHi, sayBye}; // a list of exported variables

// main.js

import {sayHi, sayBye} from './say.js';

sayHi('John'); // Hello, John!

sayBye('John'); // Bye, John!

=> lưu ý về **export default**

39

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

React Component cũng là function, nên cũng tuân theo cú pháp import/export của javascript

1. Cú pháp Import/Export

- Tách riêng component là 1 file

- export function

const MyFunction(){

return (

//todo

)

}

export default MyFunction;

---

import MyFunction from ...

2. Về cách viết code

Khi sử dụng export default và export nhiều variables/functions:

Default : export default function Button( ) { } import Button from './Button.js'; Named export function Button( ) { } import { Button } from './Button.js';

40

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#21. React JSX**

Tài liệu: https://react.dev/learn/writing-markup-with-jsx

- Component >< HTML ở chỗ, có thể code được javascript bên trong html JSX là cũ pháp giúp viết code html bên trong file javascript (.js) => .jsx 1. Convert HTML sang JSX

<h1>Hedy Lamarr's Todos</h1>

<img

src="https://i.imgur.com/yXOvdOSs.jpg"

alt="Hedy Lamarr"

class="photo"

>

<ul>

<li>Invent new traffic lights </li>

<li>Rehearse a movie scene </li>

<li>Improve the spectrum technology

</ul>

export default function TodoList() {

return (

// ???

)

}

nó không hoạt động, vì không tuân theo đúng nguyên tắc của JSX 2. Nguyên tắc của JSX

- Luôn trả ra 1 single root element

- Đóng tất cả các tags . ví dụ <input/> <img/>

- Sử dụng className, thay vì class (vì keyword class bị trùng trong file javascript)

41

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#22. Sử dụng Variables với JSX**

Tài liệu:

https://react.dev/learn/javascript-in-jsx-with-curly-braces

Javascript có các kiểu dữ liệu chính:

Nhóm 1:

string

number

Nhóm 2:

object

array

Boolean => render nothing

null/undefined => render nothing

Để sử dụng biến trong JSX, sử dụng { variable\_name } Lưu ý ngoại lệ là thuộc tính (attribute) style => sử dụng CSS inline

42

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#23. Bài tập design Component Input**

Design giao diện theo yêu cầu:

- Xóa tất cả giao diện đang có, xóa import file css (để không vỡ giao diện) - Tạo component **InputTodo**

- Viết cú pháp jsx để tạo giao diện

Các bước thực hiện:

- Tạo component mới (InputTodo)

- Import vào file **App.tsx**

Lưu ý:

**Về cách đặt tên file extension component**

**.jsx** => code javascript với React

.**tsx** => code typescript với React

.js => file javascript thông thường

.ts => file typescript thông thường

**Tên file đặt tùy thích** (viết hoa, viết thường thoải mái), **nhưng tên component phải viết hoa chữ cái đầu tiên.**

Không nhất thiết phải **import React from ‘react’** => cách nhiều khóa học cũ dạy Với react 18, khi viết code javascript trong file .jsx or tsx => không cần import React

43

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#24. Props là gì**

Tài liệu: https://react.dev/learn/passing-props-to-a-component

1. Phân loại component

- Cha (parent)

- Con (child)

2. Props

Props, là viết tắt của property (properties)

translate: thuộc tính

feeling: tài sản kế thừa từ cha => con

Props là cách thức các component giao tiếp với nhau, giúp component cha truyền dữ liệu xuống component con

Gồm 2 bước:

**1. Pass props to child component (truyền props xuống component con)** export default function Profile() {

return (

<Avatar

person={{ name: 'Lin Lanying', imageId: '1bX5QH6' }}

size={100}

/>

);

}

**2. Read props (from child component)**

MyFunction (props) {

...

}

44

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#25. Props với Typescript**

1. Tại sao cần typescript

- Được check và gợi ý code (chấm hết :v )

2. Sử dụng với Typescript

- Định nghĩa type với interface https://react-typescript

cheatsheet.netlify.app/docs/basic/getting-started/basic\_type\_example/#types or-interfaces

Tham khảo : https://hoidanit.com.vn/khoa-hoc/react-pro-typescript-thuc-hanh du-an-portfolio.html?id=6458f8f188a9e90d1174e47d

- Code typescript thực chất là javascript + định nghĩa type

3. Lưu ý khi viết code cho beginners

Do: MyFunction (props)

Don't: MyFunction({ })

45

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#26. React Devtool**

Tài liệu: https://react.dev/learn/react-developer-tools

Download: https://chrome.google.com/webstore/detail/react-developer tools/fmkadmapgofadopljbjfkapdkoienihi

46

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#27. Khái niệm Re-render** Thay đổi props (sử dụng devTool)

=> component re-render

47

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#28. Rendering Lists**

Để sử dụng vòng lặp với React bên trong JSX, chúng ta **sử dụng map** (thay vì for, while)

Về vòng lặp map: https://developer.mozilla.org/en

US/docs/Web/JavaScript/Reference/Global\_Objects/Array/map

Sử dụng map, thay vì vòng lặp for, hay while (do while) trong JSX vì: - Vòng lặp map “trả ra” một mảng mới, không làm ảnh hưởng tới giá trị cũ

- Sử dụng for, hay while, chỉ đơn thuần là “kiểm tra các phần tử”, không tạo ra mảng mới (data mới), và có thể làm ảnh hưởng tới giá trị cũ

**=> Nguyên tắc đơn giản với JSX của React, là dùng map** để lặp các phần tử (thay vì for, while)

48

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#29. Key**

Lưu ý: warning khác với error, mặc dù đều có chữ màu đỏ (chữ màu đỏ thường không tốt, right ? )

**Warning**, đa phần là không ảnh hưởng tới hệ thống, bạn fix được thì hiệu năng ứng dụng cao, không fix được cũng chẳng sao

**Error**: đa phần là lỗi cần fix (vẫn có ngoại lệ, là có lỗi xảy ra, hệ thống vẫn chạy bình thường, về ví dụ minh họa, các video tiếp theo sẽ có bạn nhé).

1. React Key

React key, là các React nhận biết các phần tử HTML trên màn hình.

React dùng key (tương tự chúng ta dùng căn cước công dân), để định danh từng phần tử HTML.

Mục đích định danh này, là mỗi khi cập nhật (quá trình re-render), **react chỉ cần cập nhật “chính xác phần tử đấy”, mà không cần phải “vẽ lại/re-render” cả trang web,** như vậy hiệu năng ứng dụng sẽ cao.

2. Tại sao không dùng Id, mà lại dùng key

Khi chúng ta sử dụng React, tư duy cần phải theo hướng “component” **Với id, trên 1 website, không thể có 2 phần tử với cùng ID, nếu có, sẽ gây ra lỗi.**

Với key và tư duy component, trên 1 website có thể tồn tại bao nhiêu component cũng được (không giới hạn số lượng).

Và nếu trùng key cũng không sao (vì React rất thông minh, nó sẽ biết key nào, ứng với component nào, chứ không đơn thuần dựa vào tên của key)

49

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 4: React Hook**

*Học kiến thức react “hiện đại” bằng cách sử dụng Hook*

**#30. Event Handler**

https://react.dev/learn/responding-to-events

Tất cả các event thông thường của HTML, React đều hỗ trợ.

Các event hay dùng nhất với React, có thể kể tới như click, change…

Lưu ý về cách code:

**Bên trong hàm event, sử dụng Arrow function**

Ví dụ:

Không code như này:

onClick={ handleClick }

Mà Code như này:

onClick={ ( ) => handleClick( ) }

Lý do: sử dụng arrow function giúp tránh các lỗi sai không cần thiết, đồng thời giúp hiệu năng cao hơn.

(chi tiết mình giải thích trong phần nâng cao của khóa React Ultimate) https://hoidanit.com.vn/khoa-hoc/react-ultimate-react-co-ban-tu-z-toi a.html?id=640b50b96cc592d780aab976

50

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#31. State là gì ?**

https://react.dev/learn/state-a-components-memory

1. Khi cách dùng thông thường là chưa đủ

- Thay đổi biến => giá trị được thay đổi => nhưng giao diện không đổi (không re-render) - Ví dụ: tạo global vars => update

2. Ví dụ về state

- State (trạng thái), dùng để hiển thị "trạng thái" của component

- State là 1 biến đặc biệt (chỉ React hiểu nó), không share giữa các component (nghĩa là, phạm vi của nó là local)

- State thay đổi => giao diện thay đổi (re-render)

51

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#32. useState Hook**

1. Why hook

Hook là cách React có thể tương tác với component (function)

Hook hỗ trợ từ **version React 16.8** => React version cũ, sẽ dùng class component (thay vì function component để viết state)

2. useState Hook

cú pháp: const [state, setState] = useState(initialValue);

3. Ví dụ về useState hook

gõ tên input => in ra giá trị

52

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#33. Bài tập về State (part 1)**

**1. Lấy giá trị input khi submit button**

- đặt state cho input

- submit button => lấy state đấy

Phân tích các bước làm:

1. đặt state

2. onchange => cập nhật state ấy

3. khi submit => lôi state ra sử dụng (vì state được lưu tại memory của component)

53

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#34. Bài tập về State (part 2)**

Logic render array, được đưa vào input component

Các bước làm:

tạo state quản lý array

khi submit btn => cập nhật array ấy

54

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#35. Passing Function từ Cha sang Con (Parent to Child)**

1.Khái niệm Parent/Child của React (cha/con)

Component nằm ngoài, bọc component khác, thì component đấy là cha

2.Cách truyền function từ cha sang con

Để truyền function từ cha sang con, tương tự như truyền “giá trị của biến/variables”, chúng ta sử dụng props

**Lưu ý có pháp viết code: chỉ truyền tên function, không có cặp dấu đóng mở ngoặc ( )** Nếu truyền thêm dấu ( ), điều này đồng nghĩa chúng ta đang thực thi function đấy. **=> Từ component con, gọi function ấy, khi đấy mới dùng dấu ( )**

55

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#36. Lift up State**

- Lift-up state, giúp các component có thể nói chuyện với nhau (thông qua component cha gần nhất)

Thực hành:

Đưa array cho component cha quản lý

truyền function update xuống cho child component

56

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#37. Vòng đời của component**

Component là "smart object". không "dump"/ngu ngốc như html.

Tương tự như "sinh vật", component có vòng đời của nó (life cycle) Vòng đời gồm: sinh ra => lớn lên => die

**Mounting** (born) : được sinh ra -> chèn html vào giao diện (cây DOM) **Update**: cập nhật giao diện sau khi đã có cây DOM

**Unmounting** (die): xóa html khỏi cây DOM

Component thể hiện vòng đời, thông qua props và state (vì state và props, giúp "hiển thị" dữ liệu của component)

=> props/state thay đổi => giao diện auto re-render

Ví dụ:

nút bấm button

57

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#38. Về React Cơ Bản**

Hiểu được props/state là gì ?

**State**: trạng thái của component. Cần lưu thông tin gì (variables, functions) của component, sử dụng state.

Về cách khai báo và sử dụng: **const [state, setState] = useState(initialValue)**

**Props**: properties - tài sản chuyển từ cha sang con. Có thể truyền (variables, functions) từ cha sang con để component con có “data” để sử dụng

Khái niệm render/re-render

State/props thay đổi => component re-render (vẽ lại giao diện mà không cần reload)

- Cập nhật state

=> thông qua useState hook. **props của child component, thực chất là state của parent component**

React fiber ? (Virtual dom)

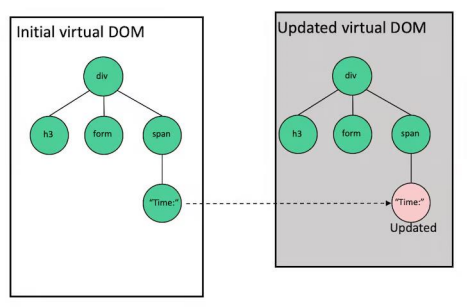
=> chỉ update những chỗ cần thiết

Batch update

https://legacy.reactjs.org/docs/faq-internals.html

58

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**59

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 5: Setup Dự Án Backend**

*Giới thiệu về project thực hành và cách cài đặt backend (được cung cấp sẵn) cho dự án frontend React.*

**#39. Giới thiệu tổng quan về dự án thực hành**

Frontend sẽ gồm 2 phần:

- React client (Vite), phụ trách quản lý CRUD của admin : localhost:5173 - React server (Next.js), normal user sử dụng : localhost:3000

Backend được cung cấp sẵn:

- Nestjs : localhost:8000

- database MongoDB, sử dụng online với mongodb atlas

Chuyện đánh bản quyền:

Điều khoản sử dụng dịch vụ khóa học của Hỏi Dân IT:

https://drive.google.com/file/d/1RR0ckOO\_OOyzh5kZ6ti0BPpOlySFn-NR/view **Tính trung thực**

**Cuộc chơi cần win-win**

60

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#40. Cài đặt Postman**

- giới thiệu sơ lược về postman

- cài đặt local: https://www.postman.com/downloads/

- import collection => chưa cần test

Link download file collection: https://drive.google.com/drive/folders/12mJ cYGhHvkySgnOPBaopk0O44hWtxTm?usp=sharing

61

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#41. Cài đặt Mongodb Compass tại local**

Link download: https://www.mongodb.com/try/download/compass

62

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#42. Setup database MongoDB Atlas**

- tạo tài khoản: https://www.mongodb.com/cloud/atlas/register - test connection với MongoDB Atlas

- lưu ý về lỗi (thỉnh thoảng ko kết nối được) => tạo cái khác

63

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#43. Kích hoạt dự án backend**

**Bước 1:** Clone dự án thực hành tại đây

Link backup download tại đây

Project cần làm trong video, có tên là : **share-create-id-react-pro-max-nextjs**

**Bước 2:** Chạy dự án (chi tiết ghi tại file README.md)

- Chạy câu lệnh : **npm i** để cài đặt các thư viện cần thiết

- Chạy dự án với câu lệnh: **npm start**

**Bước 3:** Lấy file kết quả ID

Nếu không có lỗi gì xảy ra, sẽ có message thông báo Done.

Kết quả được lưu tại file **/dist/submit-license/result.txt**

**Bước 4:** Submit kết quả

Truy cập: https://active.hoidanit.vn/license

Tại đây, bạn cần cung cấp các thông tin cần thiết, bao gồm:

- **Tên khóa học**: chọn **React Pro Max với Next.js**

- Tên tài khoản Facebook mua khóa học

- URL tài khoản Facebook mua khóa học

- Email đăng ký của học viên khi mua khóa học

cũng như **file result.txt** (có được tại bước 3)

**Nhấn submit.**

**Bước 5:** Nhận kết quả

Sau khi Submit kết quả ở trên, bạn chủ động inbox Facebook Hỏi Dân IT: https://www.facebook.com/askITwithERIC/

**Mình sẽ gửi kết quả qua email.**

Thời gian phản hồi là 24h (từ thứ 2 tới thứ 6). 1 tới 3 ngày đối với thứ 7 & chủ nhật (ngày lễ/holiday)

Làm bước 5 là thành công rồi, chuyển qua video tiếp theo các bạn nhé.

64

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#44. Run dự án backend**

Lưu ý: bạn cần xem và thực hành #43 trước khi xem video này. Cần được xác nhận kết quả submit qua email cũng như license key để thực hiện.

**Bước 1:** Clone dự án thực hành tại đây

Link backup download tại đây

Project cần làm trong video, có tên là : **share-backend-react-pro-max-nextjs**

**Bước 2:** Chạy dự án

- Cài đặt thư viện cần thiết với câu lệnh: **npm install --omit=dev** - Update file **.env:**

Cập nhật tham số cho database **MONGO\_URL**

Cập nhật license key (đã gửi trong email kích hoạt) **ACTIVE\_KEY** - Chạy dự án với câu lệnh: **npm start**

**Bước 3**: Kích hoạt bản quyền

Sử dụng kết quả nhận được email tại video #43, **cập nhật file result.txt**

**Bước 4:** Chạy lại dự án: npm start

Test dự án bằng cách:

Truy cập http://localhost:8000/

Test với postman APIs

65

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 6: Routing với React (Client Side)**

*Điều hướng trang client với React Router, đồng thời tạo trang quản trị admin để quản lý dữ liệu trong dự án thực hành*

**#45. React Router**

Tài liệu: https://reactrouter.com/en/main

1. Về cách đọc tài liệu

Chỉ có thể đọc được tài liệu cùng “main version”, không thể đọc tài liệu của chính xác version.

Có nghĩa là, có thể đọc được tài liệu chung của version 4.x, 5.x…

Nhưng không thể đọc điều tài liệu chính xác của 1 version cụ thể

=> giải pháp là nên đọc hiểu “cách dùng”, sau đấy áp dụng tương ứng cho dự án (vì công nghệ thay đổi theo thời gian)

2. React Router

What: được sử dụng để điều hướng trang mà không cần reload trang web.

How:

https://www.npmjs.com/package/react-router-dom

Cài đặt thư viện:

**npm i --save-exact react-router-dom@6.15.0**

66

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#46. Tạo Route Users**

Chưa cần design giao diện menu, quan tâm “hardcode “ route

Tài liệu:

https://reactrouter.com/en/main/start/tutorial

67

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#47. Design table User**

Các bước cần thực hiện:

- **Thư mục screens**: mỗi đường link url, là 1 screen

- **Thư mục components**: lưu trữ component được sử dụng ở các screens khác nhau (tăng tính tái sử dụng)

Yêu cầu:

Tạo component: manage.users.tsx

Render giao diện html table (hard code dữ liệu)

https://www.w3schools.com/html/tryit.asp?filename=tryhtml\_table\_intro

68

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#48. Backend và API là gì**

1. Tại sao cần backend ?

Tất cả các khóa học sử dụng “fake api” như json web server => vứt đi vì không có tính “thực tế cao”, chỉ phục vụ mục đích demo (testing)

Backend dùng miễn phí trên internet => không flexible

=> tự viết backend

**Cần backend, vì backend có khả năng kết nối tới “database”, nơi lưu trữ “dữ liệu website”**

**2. Api là gì**

Hiểu đơn giản, API là 1 đường link url.

Frontend gọi vào đường link url này để lấy dữ liệu

**API được backend viết**, Frontend sử dụng.

69

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#49. Postman để test API**

Yêu cầu:

- Đã cài đặt thành công Postman trên máy tính

- Đã có file collection và src backend (cũng như setup database)

Các bước thực hiện:

1. Run Backend

2. Run Postman (đã import collections)

3. Test APIs

Lưu ý khi chạy api có lỗi:

- Để biết được API đúng hay sai, dùng postman (với data được fill mẫu sẵn). **Nếu postman chạy được => frontend sai**

- Backend không sai. Lỗi xảy ra, là frontend "truyền sai thông tin vào api".

**Trường hợp cả postman và frontend (truyền đúng data), nhưng kết quả trả sai** => APIs sai (các bạn report mình sẽ update)

70

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#50. Gọi API với React**

1. Công cụ hỗ trợ sẵn: Fetch

https://developer.mozilla.org/en-US/docs/Web/API/Fetch\_API/Using\_Fetch Fetch là công cụ hỗ trợ sẵn của javascript để gọi API phía client.

**Với version node.js >= 18, có thể dùng fetch ở phía server**

https://nodejs.org/en/blog/announcements/v18-release-announce#fetch-experimental => dùng fetch để thống nhất giữa client và server

2. Các thư viện hỗ trợ gọi APIs khác

- **axios**: (run client và server)

https://www.npmjs.com/package/axios

(các khóa học về React khác, ví dụ như khóa React Test Fresher, mình sử dụng axios) https://hoidanit.com.vn/khoa-hoc/giai-ma-trinh-do-bai-test-fresher react.html?id=640beb7df7099c369b3bc695

Tối ưu sẵn cache:

SWR: https://www.npmjs.com/package/swr

React Query: https://tanstack.com/query/v5/docs/react/comparison

3. Test API với Fetch

- Với client (fetch trong useEffect). Why ?

- Với server (fetch ở đâu cũng được)

- Test API login

- Test API get all users (hardcode bearer token ở header)

**'Authorization': `Bearer ${token}`**

71

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#51. Mô hình stateless**

- Stateless, là hình thức xác thực người dùng thông qua token

(**access token/refresh token**).

Thông thường token sẽ được gửi kèm ở header request (hoặc cookies).

Xác thực người dùng gồm: người đấy là ai và người đấy có quyền làm gì (authentication và authorization)

Ví dụ: user là admin@gmail.com , có quyền CRUD user

- Token thường được gửi dưới định dạng JWT (**json web token**) đã được mã hóa và có tính bảo mật

- Mỗi lời gọi request được thực hiện từ Frontend tới Backend, đều cần đính kèm token.

Backend sẽ có cơ chế để "giải mã" token, từ đấy, biết được người dùng nào đang thực hiện request (authentication),

đồng thời có cho phép thực hiện hành động request hay không (authorization)

Frontend chẳng cần làm gì, ngoài việc:

1. login lấy token (lưu trữ lại, thông thường là localstorage và cookies) 2. Tất cả lời gọi request gửi lên Backend, đều cấn gán thêm token trên.

3. Trường hợp token không hợp lệ (sai định dạng, hết hạn) => backend sẽ có thông báo lỗi trả về

72

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#52. Hiển thị list User**

- Đặt biến state

- Call API -> set state

(hard code token ở header)

- Render view

test sự thay đổi = cách dùng postman add new/delete a user

73

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#53. Sử dụng key React hiệu quả**

Tài liệu: https://stackoverflow.com/questions/28329382/understanding-unique-keys for-array-children-in-react-js/43892905#43892905

1. Tại sao React cần key

key là cách react "xác định/định danh" phần tử HTML trên màn hình (tương tự ID)

Component sẽ dựa vào key để tối ưu hóa hiệu năng khi:

- Edit

- Delete

- Add new

=> chỉ có phần tử nào được edit/delete/add new thì thay đổi nơi đó. không render lại toàn bộ component (chi phí paint đắt,

dễ gây giật/lag)

2. Chuyện key = index của array

https://github.com/facebook/react/issues/1342#issuecomment-39230939

Mặc định, nếu không đặt key, React "tự động" lấy key = index

https://github.com/facebook/react/issues/1342#issuecomment-39274004

Thông thường, nếu chúng ta không modify (thêm/sửa/xóa phần tử, filter, sort... của array => thay đổi thứ tự -index).

=> đặt key = index vẫn hoạt động ok

Trường hợp đặt key = index gây ra bugs: (tương tự với random key/uuid cho key)

Nguyên tắc của key:

1. Unique (duy nhất): các key cần khác nhau (không giống nhau) => id

2. static (tĩnh): key không được thay đổi giữa các lần render. vì key thay đổi => react vẽ lại phần tử đấy (replace)

**Ví dụ về bad code:**

74

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

ví dụ **arr = ['fresher', 'junior', 'middle', 'senior' ];**

<ul>

{

arr.map((item, index) => {

return <li **key={index}**> {item} </li>

})

}

</ul>

// kết quả:

<ul>

<li **key=1**>fresher</li>

<li **key=2**>junior</li>

<li **key=3**>middle</li>

<li **key=4**>senior</li>

</ul>

Nếu thao tác "modify" (sửa đổi) mảng trên, ví dụ: **xóa phần tử 'fresher'**

=> new **arr = ['junior', 'middle', 'senior' ];**

//kết quả

<ul>

**<li key=1>junior</li>**

**<li key=2>middle</li>**

**<li key=3>senior</li>**

</ul>

=> tất cả phần tử đều thay đổi key => react cần vẽ lại toàn bộ giao diện @@

75

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

//Part 2:

Cách sử dụng key hiệu quả

1. **Sử dụng id (backend cung cấp)**

**2. Nếu không có id, cần convert dữ liệu để tạo ra id => rồi map** không dùng uuid, hay random number "bên trong vòng map"

{

arr.map((item, index) => {

return <li key={uuid()}> {item} </li>

})

}

=> dùng như trên, mỗi lần render, tất cả các phần tử đều replace (do bị thay đổi)

76

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**Chapter 7: Hướng dẫn sử dụng Chrome Dev Tool (Bổ Trợ)**

*Một chương đặc biệt hướng dẫn kỹ năng sử dụng Devtools, phục vụ đắc lực cho quá trình debug code. Chương này đặc biệt, vì rất nhiều bạn không có kỹ năng debug này.*

**#54. Thành phần của dev tool**

- Tab Elements: Xem giao diện html, code css trực tiếp (hiểu đơn giản là cây DOM)

- Console: check thông tin code: info, error, warning

error => fix khi code die, không đồng nghĩa là error => fix

warning => fix được thì càng tốt

- sources: => dùng để đặt breakpoint (quá trình debug, xem code chạy)

- Network: dùng để check api giữa frontend và backend

Lưu ý về red button (record network)

-> chọn options fetch/xhr để filter data

- Application: check local storage, session storage, cookies, indexedDB ... (data lưu tại browser)

77

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#55. Kỹ thuật check api**

Sử dụng tab network để kiểm tra api

Nhớ đã chọn records network

Chọn api, đọc thông tin của api, bao gồm:

**Tab headers** => check url , method, thông tin ở headers… **Tab payload** (nếu có): thông tin đính kèm api **Preview** (xem phản hồi dưới dạng object)

**Response**: xem phản hồi dưới dạng text

78

*Khóa Học React Pro Max với Next.js - Youtube Channel ‘Hỏi Dân IT’*

**#56. Nguyên tắc khi gọi API không có kết quả**

- check postman (với data mặc định) xem có kết quả hay ko - f12 check tab network. check url, method, data... đã chính xác chưa

- Minh họa trường hợp truyền sai data

79