

BÀI THU HOẠCH III

(Phần I)

TÌM HIỂU VỀ PWA, VIRTUAL DOM, LOCAL STORAGE

Thực tập sinh: Lê Vương Khánh

Người hướng dẫn: Nguyễn Trần Xuân Lộc, Trương Tấn Sang

MỤC LỤC

I. TÌM HIỂU VỀ PWA	2
I.1 Định nghĩa.	2
I.2 Các đặc điểm chính	3
I.3 Những điều cần biết để xây dựng 1 PWA hoàn chỉnh.	4
II. TÌM HIỂU VỀ VIRTUAL DOM.	5
II.1. Định nghĩa	5
II.2 Các đặc điểm chính của VDOM	6
II.3 So Sánh DOM và VDOM	6
III. TÌM HIỂU VỀ LOCAL STORAGE.	7
III.1. Định nghĩa	7
III.2 Các đặc điểm chính	8
III.3 Những lưu ý khi sử dụng Local Storage	9

I, TÌM HIỂU VỀ PWA

1. Định nghĩa

PWA, hoặc Progressive Web App, là một loại ứng dụng web được thiết kế để cung cấp trải nghiệm người dùng giống với ứng dụng di động truyền thống nhưng chạy trên trình duyệt web. Xu hướng về Progressive Web Apps (PWA) tiếp tục phát triển và thúc đẩy sự phổ biến của chúng trong lĩnh vực phát triển ứng dụng web trên toàn thế giới.

2. Các đặc điểm chính

Dưới đây là các đặc điểm chính của PWA:

- Khả Năng Hoạt Động Offline: PWA cho phép người dùng truy cập nội dung và chức năng của ứng dụng mà không cần kết nối internet. Điều này làm giảm sự phụ thuộc vào mạng và cải thiện trải nghiệm người dùng trong điều kiện mạng không ổn định.

- Tích Hợp Trên Màn Hình Chính: Người dùng có thể "cài đặt" PWA trực tiếp lên màn hình chính của thiết bị giống như cài đặt ứng dụng di động thông qua biểu tượng hoặc tùy chọn trình duyệt.

- Tương Thích Đa Nền Tảng: PWA có khả năng tự điều chỉnh giao diện và hiệu suất để phù hợp với các thiết bị và trình duyệt khác nhau, bao gồm máy tính, điện thoại di động và máy tính bảng.

- Tính Bảo Mật Cao: PWA sử dụng kết nối bảo mật qua HTTPS và tuân thủ các tiêu chuẩn bảo mật nghiêm ngặt, đảm bảo an toàn thông tin người dùng.

- Cập Nhật Tự Động: PWA có thể cập nhật một cách tự động mà không cần người dùng thực hiện tải về hoặc cài đặt lại từ cửa hàng ứng dụng.

- Tối ưu hóa Hiệu Suất: PWA được tối ưu hóa để có thời gian tải nhanh và hoạt động mượt mà, giúp cải thiện trải nghiệm người dùng.

- Tiết Kiệm Dung Lượng: PWA thường nhẹ hơn so với các ứng dụng di động truyền thống, giúp giảm tải lên và tải xuống.

- Sử Dụng Caching: Sử dụng Service Workers, PWA có khả năng lưu trữ cache để tăng tốc độ truy cập và giảm tải máy chủ.

- Tích hợp Thông Báo: PWA có khả năng gửi thông báo đến người dùng, cải thiện tính tương tác và cung cấp thông tin quan trọng.

- Khả năng Chia Sẻ và Lưu Trữ: Người dùng có thể dễ dàng chia sẻ liên kết tới PWA và lưu trữ ứng dụng trên thiết bị của họ mà không cần nhiều không gian lưu trữ.

Những đặc điểm này giúp PWA trở thành một lựa chọn mạnh mẽ cho việc phát triển ứng dụng web mang tính tiện ích và trải nghiệm người dùng cao cấp trên nhiều nền tảng và thiết bị.

3. Những điều cần biết để xây dựng 1 PWA hoàn chỉnh

Để xây dựng một PWA đầy đủ, ta cần biết những công nghệ:

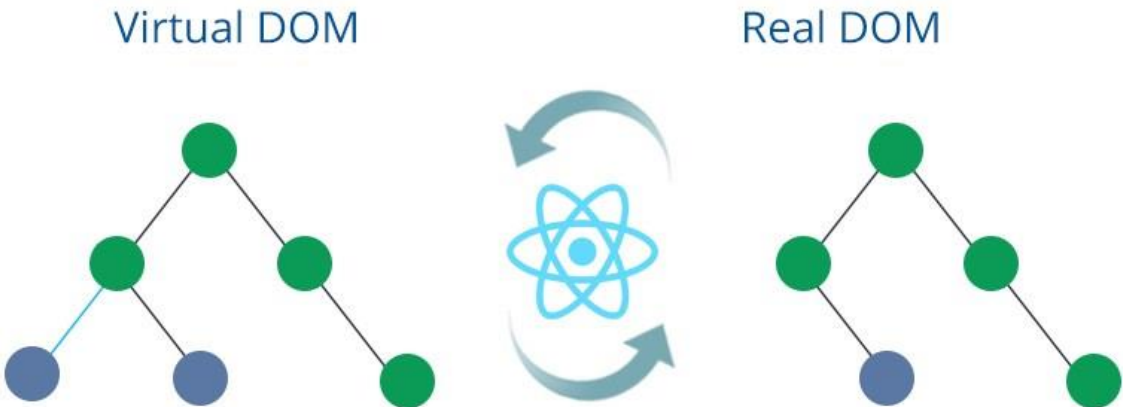
Công Nghệ	Mô Tả
HTML (HyperText Markup Language)	Ngôn ngữ đánh dấu cơ bản cho việc xây dựng giao diện người dùng của PWA.
CSS (Cascading Style Sheets)	Định dạng và tạo kiểu cho giao diện người dùng của PWA.
JavaScript	Ngôn ngữ lập trình chính cho việc thêm tính năng tương tác và xử lý logic trong PWA.
Service Workers	Công nghệ quan trọng cho việc hoạt động offline, quản lý cache và thực hiện tác vụ nền trong PWA.
Web App Manifest	Tệp JSON định nghĩa thông tin ứng dụng như tên, biểu tượng, và mô tả.
HTTPS (HyperText Transfer Protocol Secure)	Kết nối bảo mật để đảm bảo tính bảo mật của dữ liệu người dùng.
Service Worker Toolbox và Workbox	Thư viện JavaScript giúp triển khai Service Workers và quản lý cache.
IndexedDB và Web Storage	Cơ sở dữ liệu trình duyệt để lưu trữ dữ liệu PWA cho việc sử dụng offline.

Công Nghệ	Mô Tả
Fetch API	API cho phép tải và truy vấn tài nguyên từ máy chủ.
Push API	API cho phép PWA gửi thông báo đến người dùng.
Web APIs	Sử dụng các Web APIs như Geolocation API và các Web APIs khác để tích hợp các chức năng bổ sung.
Frameworks và Thư viện JavaScript	Angular, React, Vue.js và các framework khác cung cấp hỗ trợ cho việc phát triển PWA.

II. Vitural DOM

1.Định nghĩa

Virtual DOM là một khái niệm quan trọng trong lĩnh vực phát triển web, thường được sử dụng trong các thư viện và framework JavaScript như React và Vue.js. Virtual DOM là một cấu trúc dữ liệu trừu tượng được sử dụng để tối ưu hóa cập nhật DOM trong ứng dụng web, giúp cải thiện hiệu suất và hiệu quả trong quản lý giao diện người dùng.



2. Các đặc điểm chính của VDOM

Trừu tượng hóa DOM: VDOM là một cấu trúc dữ liệu trừu tượng, không phải là DOM thực tế. Nó được tạo ra để mô phỏng và đại diện cho cấu trúc của DOM thực sự trong ứng dụng.

Tối ưu hóa cập nhật: Một trong những mục tiêu chính của VDOM là tối ưu hóa quá trình cập nhật DOM. Thay vì cập nhật DOM thực sự mỗi khi có thay đổi, VDOM so sánh cây DOM ảo trước và sau thay đổi và chỉ cập nhật các phần tử thực sự cần thay đổi, giúp giảm thời gian phản hồi của ứng dụng.

Ghi lại thay đổi: VDOM thường sử dụng một quy trình ghi lại (reconciliation) để so sánh hai cây DOM ảo và xác định các thay đổi. Quá trình này bao gồm việc xác định phần tử nào cần thêm, loại bỏ hoặc cập nhật.

Hiệu suất: VDOM giúp tối ưu hóa hiệu suất bằng cách tránh cập nhật DOM thực sự một cách không cần thiết và giảm số lần tái render của trang web.

Sự kết hợp với framework: VDOM thường được sử dụng cùng với các thư viện hoặc framework JavaScript như React hoặc Vue.js. Những framework này sử dụng VDOM để quản lý cập nhật và tương tác với DOM, tạo ra một giao diện lập trình ứng dụng (API) để thực hiện các thay đổi một cách dễ dàng và hiệu quả.

Tích hợp với hệ thống UI: VDOM thường được tích hợp với hệ thống UI của framework để đảm bảo rằng các thay đổi trong trạng thái ứng dụng được phản ánh đúng cách trên giao diện người dùng.

Tương thích đa nền tảng: VDOM thường hoạt động trên nhiều nền tảng trình duyệt khác nhau và có khả năng đảm bảo tính nhất quán trong việc quản lý DOM trên các trình duyệt khác nhau.

3. So Sánh DOM và VDOM

Ta có bảng so sánh khái quát của DOM và VDOM:

Đặc Điểm	DOM	VDOM
Định nghĩa	Cấu trúc thực tế của trang	Cấu trúc dữ liệu trừu tượng
Tương tác	Tương tác trực tiếp với DOM	Không tương tác trực tiếp với DOM
Cập nhật	Cập nhật DOM thực sự	So sánh và cập nhật DOM thực sự theo cách tối ưu hóa

Đặc Điểm	DOM	VDOM
Tối ưu hóa cập nhật	Không tối ưu hóa cập nhật	Tối ưu hóa cập nhật để giảm thời gian phản hồi
Sự kết hợp với framework	Không tích hợp sẵn với framework	Thường được sử dụng cùng với framework như React
Hiệu suất	Có thể gây ra hiệu suất kém khi cập nhật thường xuyên	Cải thiện hiệu suất bằng cách tránh cập nhật không cần thiết
Ghi lại thay đổi	Không ghi lại thay đổi một cách tự động	Sử dụng quá trình ghi lại (reconciliation) để xác định thay đổi
Phạm vi tích hợp	Phạm vi hẹp, tương tác với các phần tử cụ thể trong DOM	Phạm vi lớn hơn, giúp quản lý trạng thái ứng dụng một cách hiệu quả hơn
Tích hợp với hệ thống UI	Không tích hợp sẵn với hệ thống UI	Tích hợp với hệ thống UI để cập nhật giao diện người dùng một cách đúng đắn
Tương thích đa nền tảng	Phụ thuộc vào trình duyệt và có thể gây ra sự không nhất quán	Tương thích với nhiều trình duyệt và đảm bảo sự nhất quán trong việc quản lý DOM
Sử dụng thông qua framework	Sử dụng trực tiếp trong JavaScript	Thường sử dụng thông qua framework như React

III. Local Storage

1. Định nghĩa

LocalStorage là một phần của Web Storage API trong JavaScript, được sử dụng để lưu trữ dữ liệu cục bộ (local data) trên trình duyệt web của người dùng. Nó cho phép các ứng dụng web lưu trữ thông tin trên máy tính của người dùng để có thể truy cập và sử dụng trong suốt thời gian trình duyệt đang hoạt động. Dữ liệu lưu trữ trong LocalStorage không bị xóa khi người dùng tắt trình duyệt hoặc tắt máy tính, và nó chỉ bị xóa khi người dùng thao tác xóa cụ thể hoặc thông qua mã JavaScript.

Ví dụ:

```
test.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>LocalStorage Example</title>
7  </head>
8  <body>
9      <h1>Hello, <span id="username"></span>!</h1>
10     <input type="text" id="nameInput" placeholder="Enter your name">
11     <button onclick="saveName()">Save Name</button>
12
13     <script>
14         // Hàm để lưu tên người dùng vào LocalStorage
15         function saveName() {
16             const name = document.getElementById('nameInput').value;
17             localStorage.setItem('username', name);
18             updateGreeting();
19         }
20
21         // Hàm để cập nhật lời chào với tên người dùng từ LocalStorage
22         function updateGreeting() {
23             const username = localStorage.getItem('username');
24             if (username) {
25                 document.getElementById('username').textContent = username;
26             }
27         }
28
29         // Kiểm tra xem có tên người dùng đã lưu trong LocalStorage hay chưa
30         updateGreeting();
31     </script>
32 </body>
33 </html>
34
```

2. Đặc điểm và chức năng chính

Đặc Điểm	LocalStorage
Lưu trữ cục bộ	Có
Dữ liệu dạng chuỗi	Có
Dung lượng giới hạn	Có (thường khoảng 5-10 MB)
Hết hạn tự động	Không (chỉ xóa bằng tay hoặc qua mã JavaScript)
Phạm vi trình duyệt	Có (liên kết với tên miền trang web)
Đơn giản để sử dụng	Có (cung cấp các phương thức đơn giản)
Sử dụng thường gặp	Lưu trữ cài đặt người dùng, dữ liệu phiên làm việc tạm thời, lịch sử truy cập, vv.

III. Một số lưu ý khi sử dụng Local Storage

Bảo mật: LocalStorage không cung cấp tính bảo mật cao. Dữ liệu trong LocalStorage có thể dễ dàng truy cập và thay đổi bằng cách sử dụng mã JavaScript trong trình duyệt. Do đó, nên tránh lưu trữ thông tin nhạy cảm như mật khẩu trong LocalStorage.

Hết hạn tự động: Dữ liệu trong LocalStorage không bao giờ hết hạn mà chỉ bị xóa khi bạn hoặc mã JavaScript của bạn xóa nó hoặc khi người dùng xóa thủ công thông qua cài đặt trình duyệt.

Sử dụng cẩn thận: LocalStorage thường được sử dụng để lưu trữ thông tin cài đặt người dùng, dữ liệu phiên làm việc tạm thời hoặc lịch sử truy cập. Sử dụng nó một cách cẩn thận và chỉ lưu trữ dữ liệu cần thiết.

Kiểm tra sự hỗ trợ: Trước khi sử dụng LocalStorage, hãy kiểm tra xem trình duyệt của người dùng có hỗ trợ nó không. Mặc dù hầu hết các trình duyệt hiện đại hỗ trợ LocalStorage, nhưng có thể có trình duyệt cũ hơn không hỗ trợ hoặc có giới hạn dung lượng.

Xử lý ngoại lệ: Luôn xử lý các ngoại lệ khi sử dụng LocalStorage. Ví dụ, kiểm tra xem LocalStorage đã đầy chưa trước khi thêm dữ liệu mới và xử lý các lỗi liên quan đến truy cập LocalStorage.