

Quang Le Van - Follow
Last edited by Quang Le Van on Dec 13, 2025
Contributed by

0 0 0 0

1. Lý thuyết
2. Thực hành
3. Bài tập
Expand all
Back to top
Go to bottom

Mục tiêu buổi học:
Hiểu được ReactJS là gì và lý do nên học React.
Phân biệt giữa SPA và MPA.
Cài đặt môi trường phát triển React.
Functional vs Class Component (tập trung vào functional)
Props là gì? Truyền dữ liệu giữa components
useState hook
Hiểu sự khác biệt giữa Real DOM và Virtual DOM.

1. Lý thuyết

1.1 Giới thiệu ReactJS

ReactJS là gì?

- Thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook.
- Dùng để xây dựng giao diện người dùng (UI) cho ứng dụng web.
- Dựa trên khái niệm Component-based architecture (Kiến trúc hướng thành phần).

1. MPA – Multi-Page Application

Đặc điểm:

- Mỗi khi người dùng thực hiện hành động điều hướng (như click vào link), trình duyệt sẽ gửi request mới đến server để tải toàn bộ trang HTML mới.
- Server render HTML (server-side rendering – SSR) và trả về trang hoàn chỉnh cho trình duyệt.
- Giao tiếp với server chủ yếu qua HTTP request bình thường (GET, POST...) và tải lại toàn bộ trang.

Cơ chế giao tiếp:

```
Client -- HTTP Request --> Server  
Server -- Full HTML --> Client (Page reload)
```

Ví dụ:

- Truy cập /products → trình duyệt gửi GET request đến server
- Server render trang /products.html và gửi về
- Người dùng bấm vào sản phẩm → gửi GET /product/123 → tải lại toàn trang

Ưu điểm:

- SEO tốt do HTML được server trả về đầy đủ
- Đơn giản trong triển khai ban đầu

Nhược điểm:

- Trải nghiệm không mượt: mỗi lần chuyển trang sẽ bị giật và tải lại toàn bộ
- Tốn băng thông hơn vì toàn bộ HTML, CSS, JS đều được tải lại

2. SPA – Single-Page Application

Đặc điểm:

- Ứng dụng chỉ tải một file HTML duy nhất ban đầu.
- Sau đó, điều hướng nội bộ thông qua JavaScript và React Router (hoặc Vue Router, etc).
- Giao tiếp với server chủ yếu qua AJAX (fetch / axios) và nhận về JSON.
- Giao diện không reload mà chỉ thay đổi nội dung.

Cơ chế giao tiếp:

```
Client -- fetch/axios --> Server (API JSON)  
Server -- JSON --> Client (Render bằng JS)
```

Ưu điểm:

- Trải nghiệm mượt mà: điều hướng nhanh như ứng dụng desktop
- Giảm tải cho server vì không cần render HTML liên tục
- Độ tách biệt frontend – backend (FE/BE tách rời)

Nhược điểm:

- SEO kém hơn nếu không dùng SSR hay prerendering
- Phải xử lý nhiều logic trên client (route, error, loading...)

Tóm tắt so sánh

Tiêu chí	MPA	SPA
Giao tiếp server	HTML toàn trang	JSON qua fetch / axios
Reload trang	Có, mỗi lần chuyển trang	Không, nội bộ bằng JS
Trải nghiệm người dùng	Giật, chậm hơn	Mượt, nhanh như app
SEO	Tốt	Cần cấu hình thêm
Phân tách FE/BE	Khó	Đè
Triển khai	Đơn giản	Cần toolchain (React, etc)

Tại sao dùng ReactJS?

- Dễ học, cộng đồng mạnh.
- JSX giúp viết UI dễ dàng hơn.
- Có Virtual DOM tối ưu hiệu năng.
- Tái sử dụng component dễ dàng.
- Hỗ trợ tốt SEO khi kết hợp với SSR (Next.js).

1.2 Cài đặt React App với create-react-app

Yêu cầu:

Đã cài đặt Node.js và npm (hoặc yarn).

Kiểm tra:

```
node -v  
npm -v
```

Cài đặt project React:

```
npx create-react-app hello-react  
cd hello-react  
npm start  
Mở trình duyệt: http://localhost:3000
```

1.3 Functional vs Class Component

Tiêu chí	Functional Component (Hook)	Class Component
Cú pháp	Hàm	class extends React.Component
Quản lý state	useState hook	this.state + this.setState()
Lifecycle	useEffect hook	componentDidMount,...
Ưu tiên sử dụng	✓	✗

→ ✓ Tập trung 100% vào Functional Component với hook.

1.4 State với useState

const [count, setCount] = useState(0);

- Dùng để lưu trạng thái: text, number, array, object...

• Cập nhật: setCount(count + 1)

1.5 Props & Component Composition

Props là dữ liệu được truyền từ cha → con:

```
<ProductCard name="iPhone" price={3000000} />
```

Props là read-only, có thể destructure:

```
function ProductCard({ name, price }) {  
  return <>{name} - {price} VND</>;  
}
```

1.6 Real DOM vs Virtual DOM

1. DOM thật (Real DOM) là gì?

✓ Định nghĩa:

- DOM (Document Object Model) là mô hình dữ liệu cây đại diện cho cấu trúc HTML của trang web.

• DOM thật là thứ trình duyệt tạo ra từ HTML – bạn thấy nó khi mở DevTools (F12 → tab Elements).

✗ Nhược điểm:

- DOM thật rất chậm khi thay đổi nhiều lần.

• Mỗi lần bạn thay đổi một phần nhỏ (VD: sửa nội dung thẻ h1), trình duyệt phải:

- Xác định phần tử bị thay đổi.

• Vẽ lại layout (reflow).

• Vẽ lại pixel trên màn hình (repaint).

Điều này khiến ứng dụng chậm dần nếu cập nhật nhiều hoặc liên tục.

2. DOM ảo (Virtual DOM) là gì?

✓ Định nghĩa:

- Virtual DOM là một bản sao nhẹ của Real DOM, do React tự quản lý trong bộ nhớ (RAM).

• Mỗi khi dữ liệu thay đổi, React không cập nhật Real DOM ngay lập tức, mà:

• Tạo ra một Virtual DOM mới.

• So sánh với bản Virtual DOM cũ (diffing).

• Chỉ thay đổi đúng phần cần thiết trên Real DOM.

Cơ chế này giúp React nhanh hơn rất nhiều so với thao tác trực tiếp với DOM thật.

3. Cơ chế hoạt động chi tiết:

So sánh:

```
[State/Props thay đổi]  
[React tạo Virtual DOM mới]  
[So sánh (diff) với Virtual DOM cũ]  
[Tìm ra những thay đổi (patch)]  
[Cập nhật Real DOM tối thiểu]
```

Tiêu chí	Real DOM	Virtual DOM
Là gì?	Cây DOM thật trong trình duyệt	Bản sao DOM trong bộ nhớ
Tốc độ	Chậm hơn vì thao tác trực tiếp vào DOM	Nhanh hơn nhờ so sánh (diffing)
Hiệu suất	Thấp nếu cập nhật nhiều phần tử	Cao do chỉ update phần thay đổi

Dùng ở đâu?

HTML thuần, jQuery

React, Vue, v.v.

Mình họa:

- Real DOM: Cập nhật trực tiếp vào cây → chậm.

• Virtual DOM: So sánh sự khác biệt (diff), chỉ cập nhật phần cần thiết.

2. Thực hành

2.1 Tạo project React đầu tiên

Tạo folder mới hoặc chọn nơi lưu.

Chạy:

```
npx create-react-app hello-react  
cd hello-react  
npm start  
Mở trình duyệt: http://localhost:3000
```

2.2 Render chuỗi sử dụng Template Literal và JSX

Trong App.js:

```
function App() {  
  const name = 'Quang';  
  const message = `Chào mừng ${name} đến với ReactJS!`;  
  
  return (  
    <div>  
      <h1>{message}</h1>  
    </div>  
  );  
}  
  
export default App;
```

2.3 Render thêm thông tin với JSX

```
function App() {  
  const user = {  
    name: 'Quang',  
    age: 25  
};
```

```
  return (  
    <div>  
      <h1>Chào mừng ${user.name} đến với ReactJS!</h1>  
      <p>Tuổi: ${user.age}</p>  
    </div>  
  );  
}
```

3. Bài tập

Bài tập 1: Tạo một ứng dụng ReactJS và hiển thị danh sách users dưới dạng grid (table)

Last changed by

Quang Le Van - Follow

0 0 0 0

Add a comment

Published on Try HackMD