Bai 5

React Hooks

Module: BOOTCAMP WEB-FRONTEND

Thảo luận

- **React Hooks**
- **useState hook**
- **useEffect hook**
- **Hook custom**

Mục Tiêu

- Trình bày được khái niệm Hooks
- Triển khai được useState hook
- Triển khai được useEffect hook
- Xây dựng được hook custom

React Hooks

- Hooks là một bổ sung mới trong React 16.8.
- Hooks là những hàm cho phép “kết nối” React state và lifecycle vào các components sử dụng hàm.
- Với Hooks bạn có thể sử dụng state và lifecycles mà không cần dùng ES6 Class
- Lợi ích của hook
  - Khiến các component trở nên gọn nhẹ hơn
  - Giảm đáng kể số lượng code, dễ tiếp cận
  - Cho phép chúng ta sử dụng state ngay trong function component

Why React Hooks?

Sau một thời gian làm việc với React thì có lẽ chúng ta sẽ bắt gặp một trong số các vấn đề sau:

- ❖ “Wrapper hell” các component được lồng (nested) vào nhau nhiều tạo ra một DOM tree phức tạp.
- ❖ Component quá lớn.
- ❖ Sự rắc rối của Lifecycles trong class

UseState Hook

- Là một hook cơ bản của reactjs version > 16.8.
- Giúp chúng ta có thể dùng state trong functional component.
- Input: initialState (giá trị hoặc function)
- Output: một mảng có 2 phần tử tương ứng cho state và setState.

UseState Hook

Ví dụ:

const [username, setUsername] = useState(“default name”);

Giá trị khởi tạo của username sẽ là “default name”
Chúng ta có thể sử dụng hàm setUsername để thay đổi giá trị của biến username.
setUsername(“new name”);

UseState Hook

Thực hành: useState

UseEffect

- useEffect là một hook trong React Hooks cho phép chúng ta làm việc với các life cycle ở functional component.
- useEffect Hook là của 3 phương thức componentDidMount, componentDidUpdate, và componentWillUnmount kết hợp lại với nhau.
- useEffect cho phép chúng ta xử lý các logic trong các vòng đời của component và được gọi mỗi khi có bất cứ sự thay đổi nào trong một componnet.

UseEffect

Vi dụ:

```
export const EffectDemo =()=>{
      //State
      const [fullName, setFullName] = useState({name: 'name', familyName: 'family'});
      const [title,setTitle] = useState('useEffectQ in Hooks');

      //useEffect useEffect(() => {
          console.log('useEffect has been called!'); setFullName({name:'SonMc',familyName:
          'CodeGym'});
      });

      return(
          <div>
              <hl>Title: {title}</hl>
              <h3>Name: {fullName.name}</h3>
              <h3>Family Name: {fullName.familyName}</h3>
          </div>
      );}
```

UseEffect sẽ được gọi mỗi khi component thay đổi.

UseEffect

Ví dụ:

Chúng ta cũng có thể điều khiển hàm ***useEffect*** bằng câu lệnh điều kiện, nó chính là tham số thứ 2 của hàm ***useEffect().*** Tham số thứ 2 của useEffect là một mảng, mảng này cho biết rõ chỉ gọi ***useEffect()*** khi giá trị phần tử trong mảng thay đổi. Ví dụ:

```
useEffect(() => {
  console.log('useEffect has been called!');
  setFullName({ name: 'New Name', familyName: 'CodeGym' });
}, [fullName.name]);
```

Như vậy hàm ***useEffect()*** chỉ được gọi 2 lần: 1 lần khi render components, 1 lần khi set name thành "New name".

UseEffect

Thực hành: useEffect

Hooks Custom

- Custom Hooks là những hooks mà do lập trình viên tự định nghĩa với mục đích thực hiện một chức năng nào đó, nó thường được sử dụng để chia sẻ logic giữa các components.
- React cũng định nghĩa cho chúng ta các hooks như useState, useEffect, useContext,... cho phép chúng ta làm việc dễ dàng hơn.
- Khi đặt tên một custom hooks phải có từ khóa use ở đầu, ví dụ như: useClick(), useClock(), useQuery().

Hooks Custom

- Custom Hooks là những hooks mà do lập trình viên tự định nghĩa với mục đích thực hiện một chức năng nào đó, nó thường được sử dụng để chia sẻ logic giữa các components.
- React cũng định nghĩa cho chúng ta các hooks như useState, useEffect, useContext,... cho phép chúng ta làm việc dễ dàng hơn.
- Khi đặt tên một custom hooks phải có từ khóa use ở đầu, ví dụ như: useClick(), useClock(), useQuery().

Hooks Custom

Thực hành: Hook custom

Additional hooks

useMemo

useMemo giúp ta kiểm soát việc được render dư thừa của các component con, nó khá giống với hàm shouldComponentUpdate trong LifeCycle. Bằng cách truyền vào 1 tham số thứ 2 thì chỉ khi tham số này thay đổi thì thằng useMemo mới được thực thi.

Lưu ý khi làm việc với hook

- Trong cùng một component, bạn có thể sử dụng bao nhiêu useState và useEffect tùy ý nhưng các hook này phải gọi ở trên cùng của function, không được nằm trong vòng lặp, khu vực điều kiện, hay các function con

- Nó chỉ sử dụng trong functional component

- Khi sử dụng useEffect để lấy dữ liệu, cần kiểm tra dữ liệu đã tồn tại hay chưa. Nếu không thì hàm sẽ gửi request liên tục

Tóm tắt bài học

- React Hooks là các hàm đặc biệt cho phép sử dụng các tính năng của React (mà không cần phải tạo class). Ví dụ, useState là một hook cho phép thêm React state vào function components.
- useState hook cho phép chúng ta khai báo local state trong Function Component cách mà trước để chỉ dùng cho Class Component.
- useEffect hook để quản lý vòng đời của của một component và nó phục vụ chúng ta sử dụng trong *functional component* thay vì các *lifecycle* như trước đây trong *class component.*
- Custom Hooks là những hooks mà do lập trình viên tự định nghĩa với mục đích thực hiện một chức năng nào đó