

# Giới thiệu tổng quát về Custom hooks 🚀

Presented by: Hau Nguyen | Easy Frontend

1. Custom hooks là gì?
2. Khi nào dùng custom hooks?
3. Một ví dụ đơn giản cho custom hooks: useClock()
4. Tham khảo thêm các custom hooks

## 1. Custom hooks là gì?

- Một hook do mình tự định nghĩa ra.
- Custom hook là một function hơi đặc biệt.
- Có thể sử dụng các hooks khác như useState, useEffect, ... hoặc một custom hook khác.
- Đặt tên custom hooks theo dạng: useClock(), useClickOutside(), useQuery(), ...

```
function useMagicColor() {  
  const [color, setColor] = useState('green');  
  
  useEffect(() => {  
    const intervalRef = setInterval(() => {  
      const newColor = randomColor();  
      setColor(newColor);  
    }, 2000);  
  
    return () => {  
      clearInterval(intervalRef);  
    }  
  }, [])  
  
  // Custom hooks return data instead of JSX  
  return color;  
}
```

## 2. Khi nào dùng custom hooks?

- Tách riêng biệt phần logic ra khỏi phần UI.
- Chia sẻ logic giữa các component.

```
function MagicBox() {  
  const color = useMagicColor();  
  
  return <div class="magic-box" style={{ backgroundColor: color }}></div>  
}
```

### 3. Một ví dụ đơn giản cho custom hooks: useClock()

---

## React hooks - Clock

21:54:02

21:54:02

Component hiện tại, trước khi tách logic ra một custom hook

```
function Clock() {
  const [timeString, setTimeString] = useState('');

  useEffect(() => {
    const clockInterval = setInterval(() => {
      const now = new Date();
      // HH:mm:ss
      const newTimeString = formatDate(now);

      setTimeString(newTimeString);
    }, 1000);

    return () => {
      // cleanup
      console.log('Clock cleanup');
      clearInterval(clockInterval);
    };
  }, []);

  return (
    <p style={{ fontSize: '42px' }}>{timeString}</p>
  );
}
```

Sau khi chuyển sang custom hooks! Trời!!! Nó y chang nhau luôn 🤔 Khác mỗi cái **return**

```
function useClock() {
  const [timeString, setTimeString] = useState('');

  useEffect(() => {
    const clockInterval = setInterval(() => {
      const now = new Date();
      // HH:mm:ss
      const newTimeString = formatDate(now);

      setTimeString(newTimeString);
    }, 1000);

    return () => {
      // cleanup
      console.log('Clock cleanup');
      clearInterval(clockInterval);
    };
  }, []);

  return { timeString };
}
```

Và sử dụng trong component Clock như sau:

```
function Clock() {  
  const { timeString } = useClock();  
  
  return (  
    <p style={{ fontSize: '42px' }}>{timeString}</p>  
  );  
}
```

Một component khác cũng dùng useClock() hooks:

```
function BetterClock() {  
  const { timeString } = useClock();  
  
  return (  
    <div className="better-clock">  
      <p className="better-clock__time">{timeString}</p>  
    </div>  
  );  
}
```

## 4. Tham khảo thêm các custom hooks

- Tài liệu chính thức: <https://reactjs.org/docs/hooks-custom.html>
- Trang use hooks: <https://usehooks.com/>
- Awesome react hooks: <https://github.com/rehooks/awesome-react-hooks>
- Beautiful react hooks: <https://github.com/beautifulinteractions/beautiful-react-hooks>
- The platform: <https://github.com/jaredpalmer/the-platform>
- ESLint React hooks: <https://www.npmjs.com/package/eslint-plugin-react-hooks#installation>



Các bạn nhớ nè

- Dùng custom hooks tách logic ra khỏi UI để tái sử dụng.
- Custom hooks là một function đặc biệt sử dụng được hooks khác.
- Custom hooks return data thay vì JSX như component.

Cảm ơn các bạn đã theo dõi video của mình! ❤️

Hẹn gặp các bạn ở video tiếp theo để code cùng các bạn một custom hook đơn giản nhé! 😊