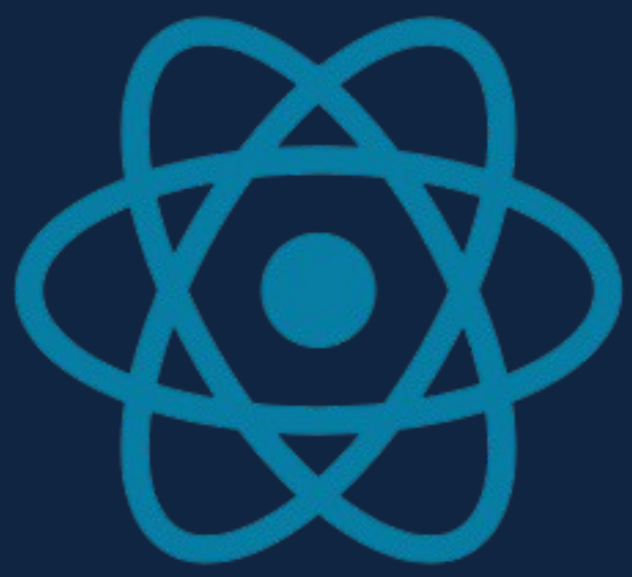
Zohaib Asghar

React Custom Hooks

How to Build Reusable Logic



What Are Custom Hooks?

Custom Hooksletyouextract and reuselogic fromcomponents.

They’re just normal functions that use React Hooks!

🧠 Rule: Must start with **use** (e.g., **useFetch**, **useToggle**)

📦 Benefit: Cleaner, reusable, and easier to test.

Why Use Custom Hooks?

✅ Keepscomponentsclean

✅ Encourages reusability

✅ Improves testability

✅ Easier to share logic across your app

Example 1: useToggle()

Simplify toggling state between true and false.

REACT

```react
import { useState } from "react";

function useToggle(initial = false) {
  const [value, setValue] = useState(initial);
  const toggle = () => setValue((v) => !v);
  return [value, toggle];
}

// Usage
const [isOpen, toggleOpen] = useToggle();
```

Example 2: useFetch()

Reusable hook for API calls 👨‍💻

REACT

```react
import { useState, useEffect } from "react";

function useFetch(url) {
  const [data, setData] = useState(null);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    fetch(url)
      .then((res) => res.json())
      .then((data) => setData(data))
      .finally(() => setLoading(false));
  }, [url]);

  return { data, loading };
}
```

Example 3: useWindowWidth()

Track window width easily 📏

REACT

```
import { useState, useEffect } from "react";

function useWindowWidth() {
  const [width, setWidth] = useState(window.innerWidth);

  useEffect(() => {
    const handleResize = () => setWidth(window.innerWidth);
    window.addEventListener("resize", handleResize);
    return () => window.removeEventListener("resize",
handleResize);
  }, []);

  return width;
}
```

Did you find it Useful?

Leave a comment!

FOLLOW

Zohaib Asghar

Like Comment Repost