

React

Abyan Majid, 2023

React Hooks	
Hook	Explanation
useState	<p>Used to save the current state of something that may change, and provide an updating function to recall the component function and therefore save the new state of the changed thing.</p> <p>Initialize State: <code>const [state, setState] = useState(initialValue);</code> Update State: <code>setState(newValue);</code> or <code>setState(prevState => newState);</code> for dependent updates.</p> <p>Example:</p> <pre>import { useState } from "react"; const [count, setCount] = useState(0); // Initialize setCount(1); // Update setCount(prevCount => prevCount + 1); // Increment</pre>
useEffect	<p>Used to handle code that produces side-effects. It runs with every render of the component, and every time a dependency has been changed.</p> <p>Example (handling API fetch and setState updating function infinite loop):</p> <pre>import { useEffect } from "react"; useEffect(() => { async function fetchPosts() { const response = await fetch("http://localhost:8080/posts"); const resData = await response.json(); // Context: setPosts is useState updating function setPosts(resData.posts); } fetchPosts(); }, []);</pre>
useNavigate	<p>Used to programmatically redirect user to another route</p> <p>Example:</p> <pre>import { useNavigate } from "react-router-dom"; const navigate = useNavigate(); function closeHandler() { navigate("/somepath"); // supports relative paths (like "..") }</pre>
useLoaderData	<p>Gets loader data which was loaded into the component in the list of routes in main.jsx or an ancestor component. <i>In other words</i>, the loader data is accessible by the route to which it was loaded, as well as every other children route.</p>

Example:

```
// main.jsx
```

```
const router = createBrowserRouter([
  {
    path: "/",
    element: <MyComponent />,
    loader: loaderFunction,
    children: [/* Children routes */] // loader data will also be
    accessible within the children routes.
  }
]);
```

```
// MyComponent.jsx
```

```
export async function loaderFunction() {
  const response = await fetch("http://localhost:8080/path");
  const resData = await response.json();
  return resData.data;
}
```

```
// MyComponent.jsx or someChildrenComponent.jsx
```

```
import { useLoaderData } from "react-router-dom";
const loaderData = useLoaderData();
```
