

Tips

React 18

Introduce new

Hooks.

@umershujaDev

List of Hooks

These are some hooks introduced in React 18

1. `useId`
2. `useTransition`
3. `useDeferredValue`
4. `useSyncExternalStore`
5. `useInsertionEffect`

1) **useId**

useId is a new hook that avoids hydration mismatches while producing unique IDs on both the client and server. It's most beneficial for component libraries that need to integrate with accessibility APIs that require unique IDs.

This fixes a problem that existed in React 17 and earlier, but it's even more critical in React 18 due to the new streaming server renderer's out-of-order HTML delivery.

2) **useTransition**

`startTransition` and **`useTransition`** consider state updates as not urgent. So, by default, other state updates are regarded as urgent. React will allow non-urgent state updates.

For example, rendering a list of search results to be interrupted by urgent state updates, for example, changing a text input.

3) **useDeferredValue**

You can use **useDeferredValue** to defer re-rendering a non-urgent component of the tree. It works similarly to debouncing, although it offers a few advantages. React will attempt the deferred render directly after the initial render is reflected on the screen because there is no predetermined time delay. The deferred render is interruptible and does not prevent the user from making a selection.

4) **useSyncExternalStore**

The new hook **useSyncExternalStore** forces synchronous updates to external stores, allowing for concurrent reads. It eliminates the requirement for `useEffect` when creating external data subscriptions and is recommended for any library that integrates with a state outside of React.

5) **useInsertionEffect**

The **useInsertionEffect** hook helps CSS-in-JS libraries to handle performance difficulties with injecting styles in the render. It's not needed to use until there's already a CSS-in-JS library. This hook will execute once the DOM is changed, but before the layout effects have had time to read the new layout. This fixes a problem that existed in previous React version 18 but is now even more critical in React 18 because React yields to the browser during concurrent rendering, allowing it to recalculate layout.

**Was this article
helpful to you?**

www.linkedin.com/in/umershujadev/