

What's New in React 18



Williams Godsfavour

Some exciting new improvements have been launched with React 18. React 18 comes with a few breaking changes, depending on how you use it. But all in all, it also brings out-of-the-box performance improvements including batching more by default, which removes the need to manually batch updates in application or library code.

Breaking Changes in **React 18**



1. Render to `createRoot`



Williams Godsfavour

```
// Before

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
  document.getElementById('root'));

// After

const root =
  ReactDOM.createRoot(document.getElementById('root'));

root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

Breaking Changes in React 18

2. Hydrate to HydrateRoot



Williams Godsfavour

```
// Before
import { hydrate } from "react-dom";

const container = document.getElementById("app");

hydrate(<App tab="home" />, container);
```

```
// After
```

```
import { hydrateRoot } from "react-dom/client";

const container = document.getElementById("app");

const root = hydrateRoot(container, <App tab="home" />);
```

Breaking Changes in React 18

3. AutomaticBatching in React 18



Williams Godsfavour

```
// Before: only React events were batched
setTimeout(() => {
  setSize((oldSize) => oldSize + 1);
  setOpen((oldOpen) => !oldOpen);
  // React will render twice, once for each state update (no
  // batching)
}, 1000);

// After: updates inside of timeouts, promises,
// native event handlers or any other event are batched
setTimeout(() => {
  setSize((oldSize) => oldSize + 1);
  setOpen((oldOpen) => !oldOpen);
  // React will only re-render once at the end (that is batching)
}, 1000);
```

Breaking Changes in **React 18**



4. Suspense On the Server



Williams Godsfavour

```
<Suspense fallback={<h1>Loading profile...</h1>}>
  <ProfileDetails />
  <Suspense fallback={<h1>Loading posts...</h1>}>
    <ProfileTimeline />
  </Suspense>
</Suspense>
```

Breaking Changes in **React 18**



5. 5 New Hooks in React 18



Williams Godsfavour

With React 18, we have five new hooks:

- `useId`
- `useTransition`
- `useDeferredValue`
- `useSyncExternalStore`
- `useInsertionEffect`

Breaking Changes in **React 18**



6. Components Can Now Render `undefined`



Williams Godsfavour

React no longer throws an error if you return `undefined` from a component. The allowed component returns values consistent with allowed values in the middle of a component tree. The React team suggests using a linter to prevent mistakes like forgetting a `return` statement before JSX.

Breaking Changes in **React 18**



7. Improved Memory Usage



Williams Godsfavour

React now cleans up more internal fields on unmount, so the impact from unfixed memory leaks in your application code is less severe. It would be interesting to see how memory usage drops compared to the previous versions.

Breaking Changes in **React 18**



8. No `setState` Warning on Unmounted Components



Williams Godsfavour

Previously, React warned about memory leaks when you called `setState` on an unmounted component. This warning was added for subscriptions, but people primarily ran into it in scenarios where the setting state was fine, and workarounds would worsen the code.