Getting Started with React Redux

[React Redux](#) is the official [React](#) UI bindings layer for [Redux](#). It lets your React components read data from a Redux store, and dispatch actions to the store to update state.

Installation[#](#)

React Redux 7.1+ requires **React 16.8.3 or later**, in order to make use of React Hooks.

Using Create React App[#](#)

The recommended way to start new apps with React and Redux is by using the [official Redux+JS template](#) or [Redux+TS template](#) for [Create React App](#), which takes advantage of **Redux Toolkit** and React Redux's integration with React components.

```
# Redux + Plain JS template
npx create-react-app my-app --template redux

# Redux + TypeScript template
npx create-react-app my-app --template redux-typescript
```

An Existing React App[#](#)

To use React Redux with your React app, install it as a dependency:

```
# If you use npm:
npm install react-redux

# Or if you use Yarn:
yarn add react-redux
```

You'll also need to [install Redux](#) and [set up a Redux store](#) in your app.

If you are using TypeScript, the React Redux types are maintained separately in DefinitelyTyped, but included as a dependency of the `react-redux` package, so they should be installed automatically. If you still need to install them manually, run:

```
npm install @types/react-redux                    Copy
```

API Overview[#](#)

Provider[#](#)

React Redux includes a `<Provider />` component, which makes the Redux store available to the rest of your app:

```javascript
import React from 'react'
import ReactDOM from 'react-dom'

import { Provider } from 'react-redux'
import store from './store'

import App from './App'

const rootElement = document.getElementById('root')
ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>,
  rootElement
)
```

Hooks[#](#)

React Redux provides a pair of custom React hooks that allow your React components to interact with the Redux store.

`useSelector` reads a value from the store state and subscribes to updates, while `useDispatch` returns the store's `dispatch` method to let you dispatch actions.

```jsx
import React from 'react'
import { useSelector, useDispatch } from 'react-redux'
import {
  decrement,
  increment,
  incrementByAmount,
  incrementAsync,
  selectCount,
} from './counterSlice'
import styles from './Counter.module.css'

export function Counter() {
  const count = useSelector(selectCount)
  const dispatch = useDispatch()

  return (
    <div>
      <div className={styles.row}>
        <button
          className={styles.button}
          aria-label="Increment value"
          onClick={() => dispatch(increment())}
        >
          +
        </button>
        <span className={styles.value}>{count}</span>
        <button
          className={styles.button}
          aria-label="Decrement value"
          onClick={() => dispatch(decrement())}
        >
          -
        </button>
      </div>
      {/* omit additional rendering output here */}
    </div>
  )
}
```

import React from 'react'
import { useSelector, useDispatch } from 'react-redux'
import {
  decrement,
  increment,
  incrementByAmount,
  incrementAsync,
  selectCount,

```
} from './counterSlice'
import styles from './Counter.module.css'

export function Counter() {
  const count = useSelector(selectCount)
  const dispatch = useDispatch()

  return (
    <div>
      <div className={styles.row}>
        <button
          className={styles.button}
          aria-label="Increment value"
          onClick={() => dispatch(increment())}
        >
          +
        </button>
        <span className={styles.value}>{count}</span>
        <button
          className={styles.button}
          aria-label="Decrement value"
          onClick={() => dispatch(decrement())}
        >
          -
        </button>
      </div>
      {/* omit additional rendering output here */}
    </div>
  )
}
```

Learning React Redux[#](#)

Learn Modern Redux Livestream[#](#)

Redux maintainer Mark Erikson appeared on the "Learn with Jason" show to explain how we recommend using Redux today. The show includes a live-coded example app that shows how to use Redux Toolkit and React-Redux hooks with Typescript, as well as the new RTK Query data fetching APIs.

See [the "Learn Modern Redux" show notes page](#) for a transcript and links to the example app source.

Help and Discussion[#](#)

The **[#redux channel](#)** of the **[Reactiflux Discord community](#)** is our official resource for all questions related to learning and using Redux. Reactiflux is a great place to hang out, ask questions, and learn - come join us!

You can also ask questions on [Stack Overflow](#) using the **[#redux tag](#)**.