



What's New In Redux



New Hooks

- ◆ useSelector
- ◆ useDispatcher
- ◆ Reselctor library



useSelector vs MapStateToProps

When an action is dispatched, `useSelector()` will do a reference comparison of the previous selector result value and the current result value. If they are different, the component will be forced to re-render. If they are the same, the component will not re-render.

With `mapState`, all individual fields were returned in a combined object. It didn't matter if the return object was a new reference or not - `connect()` just compared the individual fields. With `useSelector()`, returning a new object every time will always force a re-render by default. If you want to retrieve multiple values from the store, you can:

- ❖ Call `useSelector()` multiple times, with each call returning a single field value
- ❖ Use `Reselect` or a similar library to create a memoized selector that returns multiple values in one object, but only returns a new object when one of the values has changed.
- ❖ Use the `shallowEqual` function from `React-Redux` as the `equalityFn` argument to `useSelector()`



Reselect library

Reselect provides a function `createSelector` for creating memoized selectors. `createSelector` takes an array of input-selectors and a transform function as its arguments. If the Redux state tree is mutated in a way that causes the value of an input-selector to change, the selector will call its transform function with the values of the input-selectors as arguments and return the result. If the values of the input-selectors are the same as the previous call to the selector, it will return the previously computed value instead of calling the transform function.

Having your results memoized means that when an action is dispatched that changes something in state that is not relevant to your component then your component won't re render. This can improve performance and prevent unneeded re renders but does cost a little as well. Each item creates a `selectItemById` function and that function is a selector created with `reselect` so it's 2 memoized curried functions that take time to be created and take up memory.



Reference Links

<https://github.com/reduxjs/reselect>

<https://react-redux.js.org/api/hooks>

<https://betterprogramming.pub/increase-your-react-redux-application-performance-with-reselect-library-3f4d632a08c5>