useSelector and useDispatch

Here useSelector is similar to mapStateToProp and useDispatch is similar to mapDispatchToProp.

This way, we can insert both the current state and the action in the reducer function and the logic should look something like this.



Until now, all of our code was very similar to using React and Redux with no hooks, but now things are going to start changing a little bit and we are going to take away some of the code abstraction of writing Redux code and make possible to read and dispatch (send) data from the store a lot easier from any functional component.

Let’s start by importing ****useSelector**** from the ****react-redux**** library. ****useSelector****is a function that takes the current state as an argument and returns whatever data you want from it. It’s very similiar to ****mapStateToProps()****and it allows you to store the return values inside a variable within the scope of you functional components instead of passing down as props.



Now let’s try to dispatch an action to the store by importing ****useDispatch**** and implementing it to the scope of the functional component.



Similar to ****useSelector****, ****useDispatch**** is a function that we import from ****react-redux**** library and assign it to a variable. And with this, we are able to dispatch any action to the store by simply adding an action as an argument to the new variable like the code above and that’s it.

You are set up to start using this syntax to every functional component that needs to either read data from the state or dispatch an action to it without having to write the extra amount of code when dealing with class components.