**Lesson 8 Review**

SEND FEEDBACK

[**This**](https://github.com/udacity/reactnd-UdaciFitness-complete/tree/a3e78fe08ac785e6ce372ed37bfb5bcfa19851e3) is where we left off in the last lesson.

Redux was hooked into our app by doing:

<Provider store={createStore(reducer)}>

<View>

<AddEntry />

</View>

</Provider>

AddEntry component is connected to the Redux store:

**export** **default** connect(mapStateToProps)(AddEntry)

Now, this component will have access to dispatch.

Do you remember which other arguments the connect() function accepts? If not, please review [**this documentation**](https://github.com/reduxjs/react-redux/blob/master/docs/api.md).

The [**Redux documentation**](https://github.com/reduxjs/react-redux/blob/master/docs/api.md) provides an excellent explanation of mapStateToProps and mapDispatchToProps:

mapStateToProps - If this argument is specified, the new component will subscribe to Redux store updates. This means that any time the store is updated, mapStateToProps will be called. The results of mapStateToProps must be a plain object, which will be merged into the component’s props. If you don't want to subscribe to store updates, pass null or undefined in place of mapStateToProps.

mapDispatchToProps - If an object is passed, each function inside it is assumed to be a Redux action creator. An object with the same function names, but with every action creator wrapped into a dispatch call so they may be invoked directly, will be merged into the component’s props. If a function is passed, it will be given dispatch as the first parameter. It’s up to you to return an object that somehow uses dispatch to bind action creators in your own way. (Tip: you may use the [**bindActionCreators()**](https://redux.js.org/api-reference/bindactioncreators) helper from Redux.)

Remember that the signature of the mapStateToProps function is:

mapStateToProps(state, [ownProps])

* state is the state inside the store
* ownProps are the properties that have been passed to this component from a parent component