Selectors

In our context, ***selectors are nothing but functions which can compute or retrive data from the store***. We usually fetch the state data using mapStateToProps function like this.

**const** mapStateToProps = (state) => {

**return** {

activeData: getActiveData(state.someData, state.isActive)

}

}

Where getActiveData will be a function which returns all the records from someData having the status as isActive .

The drawback with this function is, whenever any part of the state state updates, this function will recalculate this data. In this case the calculation might be a very simple filter function. But what if this calculation takes 2-3 seconds every single time when the redux store gets updated. That too neither of the state’s data and isActive doesn’t change. This is where **Reselect** comes to rescue. What Reselect does is unless the input parameters to the function doesn’t change, re calculation doesn’t happen.

How to write a ‘reselect’ selector ?

When we use Reselect it caches the input arguments to the memoized function. So only when the arguments of the function changes from the previous call, the selector recalculates.

For a above mapStateToProps function a normal selector would be written as

**const** getActiveData = (someData, isActive) {

// returns by filtering out all someData for which isActive set to true

}

The same with reselect will be

**const** getActiveData = createSelector([someData, isActive], (data, active) => {

// callback is called only when someData or isActive changes

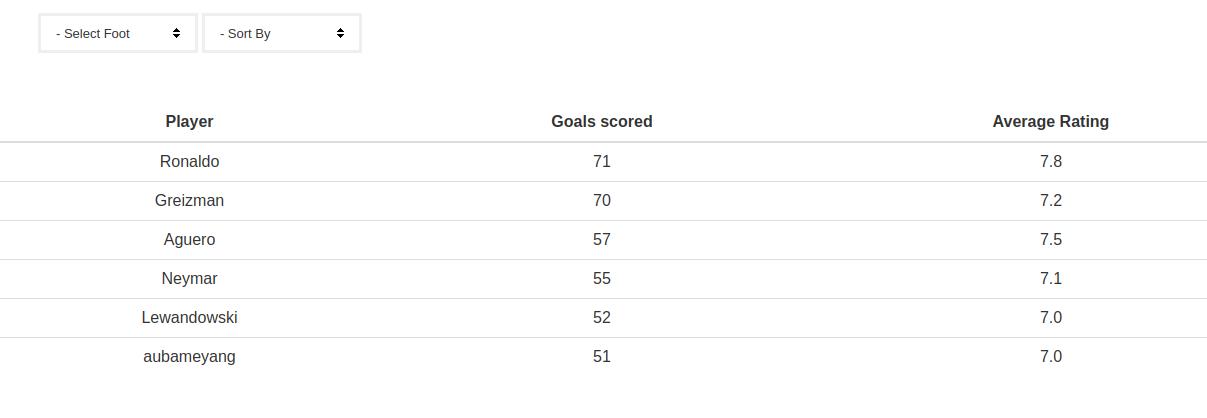
}

Reselect provides a createSelector API for creating memoized selectors. createSelector takes an array of input-selectors and a transform function as its arguments. If the values of the input selectors are the same as the previous call to the selector, it will return the previously computed value.

A *memoized selector* can itself be an input-selector to another memoized selector. *This is a very useful technique since we can chain selectors so that if any part of the chain re calculates only selectors below will it re calculate.*

React Application using reslect performance with an example

Assume that there is a list of football player’s statistics displayed in the UI like this.

[](https://codebrahma.com/wp-content/uploads/2017/09/optimizing-react-redux-application.jpg)