**My Weather Application – using Redux & calling an external API**

**Difficulties and Distractions:**

Yes I had distractions taking time away in the wine country to work on this but I used Facebook’s create-react app boilerplate which I googled and got something done, not pretty though. The difficulty yes was with Heroku for sure……

**What this Weather App does technically:**

The App searches for a given search term, passing that value to the external

OpenWeather application, along with an API key and a measurement unit. The search

begins a Redux store change, changing a few properties (isSearching, data, error);

isSearching allows a loading text to be displayed or hidden. data is the successful

response from the server and error is the error from the unsuccessful response.

These properties are mutated and available using the weather actions and weather reducers. These properties are available by connecting the React-Redux store state to the components within the application, using the connect() method!

The search begins by routing from either from the Home component and clicking on the available button, or by routing to the /search URL. This then mounts the Search component where a Form component displays the input and search button. The search value is stored in the Search components state and the search button, accesses the Search components state.term value, to begin firing the Redux action of searching with the API using that state.term value.

As mentioned, the successful response or the error response from the server is stored in the Redux store properties of data and error and are available to any connected components. Specifically, the Search component connects to the Redux store, accessing those properties and passes those properties to the Weather component that places them on the screen and displays the current weather.

**What I wish I could have completed:**

**Extra Messages “Search Term” and “Error”:**

In case there is an error when searching, it won’t display an error message at the moment, same if you try to search without having any search terms, there's

no message displaying something that could say "Please enter a search term".

That would be implemented by checking the value of props.term and if it's an

empty string onClick, it would display a message. Specifically, if

the Redux store state isSearching is true and props.term is an empty string

than display the "Please enter a search term" message, otherwise don't display

it.

**Styling:**

The styling could have been done better and is minimal and more could have been done for the app header and the Search component.

**A couple more things:**

Only the current weather is visible and providing the next 5 day forecast

would be more fun to create.

Also the Redux type's should be separated into their own types.js file and imported,

as it is the best practice with React-Redux.