React Notes

Code

Link: https://seagrant-dev.ncsa.illinois.edu/gd3/

Repository: https://opensource.ncsa.illinois.edu/bitbucket/projects/GEOD/repos/geodashboard-v3/browse

React Resources

- ES6 Imports
- rd3, demo

IDE: Webstorm (IDE for Clowder: Intellij)

- React Component (component/sensor.js)
  The component extends Component. It inherits the render method, and it returns a div. Note: The render method should only have one kid.
- The way components communicate with each other is with the redux store

- The list of the data sources and parameters come from the api (seagrant-dev.ncsa.illinois.edu/clowder/)
- To add the time to the redux state
  - You always start with an action. In the event, you can define any parameter you want as part of the event.
  - When you select the datapoint it triggers a 'Add search datasource' event. (It is defined in actions/index.js)
  - You send an event to the store. This is an event, do something with it.
  - Now, you tell the store what to do with the action. That is in the reducers. (reducers/selectedDataSources.js)
  - In selectedDataSources you tell it what you are going to do with the new data. There is a case for 'Add search datasource', In this case, it will overwrite the data_sources with the action that was sent in.
  - In Yan's case, she will add an action and a reducer.
  - Then, you have to map the state or props of the component to listen to the redux stores. This is done in the containers (containers /Sensors.js)
  - The html from each of the sensors in the explore page comes from component/sensor.js
  - The list of sensors comes from component/sensors.js
  - Each component can be represented as a tag. When in sensors.js you add sensors you can do <Sensor/> for each one.
  - The container wires the component in components folder with the react store.
  - In the containers ownProps is the props of the component.
  - export Default makes the value available to be used by other js files. Is the equivalent of making it public in other languages.
  - The function in containers/Sensors.js says in my local take whatever is in the state.sensors.data and add it to my props.sensorData Map What is on the redux store into a field in my props.
  - Whenever the redux-state updates the properties you are listening to in the container, the component re renders with the updated State
- Components - Are plain React components. They only change the local state of the components. The component has a state, but it also syncs up with the react store. For the download button, the data source component can't tell the download button what is selected. The download button needs to talk to the react store.
- Containers - Are plain react component + code that maps the state of the component to the state of the React Store.