As a beginner, I had a very difficult time adjusting to OED's development process. It was the first time I used github's branching feature, learned a new language in less than a week, and downloaded software I didn't fully understand.

My first issue was a task that involved adding a dropdown menu to the header. I originally thought that this would be easy, but the task turned out to be very difficult as I am still not completely sure what a uselector even is! Furthermore, you may quickly find that method you decided upon (to implement/modify your feature) will fall apart fast. This is because OED's webpage is implemented using BOTH typescript files with only classes (that don't allow uselectors), and typescript files with functions (that use react hooks / uselectors). As such this guide is dedicated to help you understand all the moving parts required to change/implement your desired feature.

Codebase:

- /src/client/app/components (https://github.com/OpenEnergyDashboard/OED/tree/development/src/client/app/components) contains many typescript files that render different parts of the webpage.
- At the time of creating this document /src/client/app/components/ChartDataSelectComponent.tsx is an example of one of the updated tsx files that actually utilizes a uselect.
- At the time of creating this document, /src/client/app/components/HeaderButtonsComponents.tsx is an example of a class implementation that needs to be updated to use uselector
- /src/client/app/utils (https://github.com/OpenEnergyDashboard/OED/tree/development/src/client/app/utils) contains helper functions that may be used in components. For example getPage.ts is a function that returns the name of a page

Docker:

- https://docs.docker.com/get-started/overview/
  - Docker's overview of what docker is
- https://www.youtube.com/watch?v=rOTqprHv1YE
  - Docker provides a space that 'emulates' an operating system without actually emulating it (like a virtual machine)

React Hooks:

- https://www.youtube.com/watch?v=oKvF8jcSwIg
  - This video does a good job of explaining how to implement a uselector.
  - The key takeaway is that the uselector accepts the "state" of an object as its argument, and returns a value based off that state
- https://reactjs.org/docs/hooks-intro.html
  - This article explains the where/what/when/**why**/how of uselectors