Production Calculator

Scenario One

Logged-out user starts with empty production chain, adds multiple inputs, removes an input, and clears the chain

1. (The user navigates to the production calculator tab)

Realization:

- Only the logged-out menus will be displayed; this will happen via the menus binding their v-if/shows to the state of the user system

2. (The production calculator is loaded in)

Realization:

- The method to display the loaded factory is called; this could be done through lifecycle hooks?
    - The loaded factory object is fetched, and the visual output is generated
        - Will need to generate visual output for each node
            - Includes their respective IRPTU and crafter data
        - Will need to refresh menus
        - Will need to refresh local lists of objects, such as a possible input item list
        - Will need lots of logic here that I don't quite understand yet
- The loaded factory store will start empty, so none of the above actions will have a noticeable result

3. (The user enters the ID of the input they wish to add)

Realization

- The user chooses from a list of pre-loaded item IDs
    - I'll have to decide how fancy I want to make this menu
- How will the menu have access to the list of valid IDs?

4. (The user enters the amount of the item required

Realization

- The input will be filtered for validity

5. (The user enters the time unit of the request)

Realization

- The user chooses from a list of preloaded time units (minutes, seconds, hours, etc)

6. (The user submits the request)

Realization

- All the data from the request is fetched; I imagine I can collect all the data using a form, and then pass it to some method as opposed to calling an HTTPS method (this is what I do with the user forms)
- A method in the loaded factory store for adding an input to the system will be called
- The loaded factory store will then make a call to what is currently the “Output” module; this module will be responsible for validating the passed data, as well as making the proper calls to the calculator module in the correct order
    - What happens if the data is invalid? The thing being returned from output is the object, so how will the loaded factory store know the request to update the object didn’t work
- The output module will validate the updated object
- The output module will then return the updated object
- The loaded factory object will then be swapped out with the new, updated object
- Finally, the Reload method of the ProductionCalculatorView will be called

ProductionCalculatorView → loadedFactory → Output → Calculators

Calculators → Output → loadedFactory → ProductionCalculatorView

7. (The user submits some more requests)

8. (The user wishes to remove x amount of an input from the factory)

Realization

- Similar to the IRPTU addition request, the data will be collected using a form (if not, I'll just find some other way to do it; it's not a big deal)
    - The inputs could also be displayed separately from the nodes of the factory, and they'd be editable in that case; this is similar to the website I reference a lot
    - I wouldn't have to change the underlying removal logic in order to facilitate this
    - I'm just going with the first option for now
- The data will be input using a selection control containing the IDs of the inputs
- The amount to remove will be bounded the IRPTU of the item
- I could have an internal state here that prevents the second input from being entered until the first input has been entered
- How will the menu have access to the input data