COMP6080: Web Front-End Programming

Tutorial 4

Sidney Pham
`sidney.pham@unsw.edu.au`

UNSW

Week 4, Term 3 2020

Exercise: A Simple Form

Build a simple form that collects an email address and first name from a user.

Valid inputs:

- A valid first name is defined as only lowercase and uppercase letters, between 2 and 20 characters long inclusively.
- A valid email is defined by the regular expression `/^.+\@.+\..+$/` (read more on that here).

Important facts about form state:

- The submit button remains disabled until both a valid first name and valid email are entered. The state change occurs on `keyup` of either input fields.
- On `blur` of either input fields, if their value is invalid, the background of the input is turned a light red.
- On `focus` of either input fields, we remove any error backgrounds.

Part 1: In Vanilla JS

Let's first do it using Vanilla JS.

Part 2: In React

Let's see how we'd do this in React. There's a useful but simple tutorial on the official docs here. Note that it shows you class components, whereas we learn functional components in this course. I would recommend going through the MAIN CONCEPTS and some of the HOOKS sections.

Controlled Components

The main idea is to make the React state be the ‘single source of truth’. That is, the `value` is controlled by React, as well as other handlers like `onSubmit` and `onChange`. Really, I’d just read the tutorial.

Part 3: Formik

When you're tasked with making a lot of forms, this can get tedious. More complete solutions exist, and the one I use is called Formik. Let's see how it works.

Exercise: Basic React State

Use a basic HTML table to build a very simple ‘Tic-Tac-Toe’ game. Users can take turns clicking on cells, filling them with `X` or `O` depending on whose turn it is. Do this using React.

As bonus exercises you may want to explore:

- Using loops as part of rendering.
- Accounting for the ‘finish’ (goal) state.

Bonus: The Solution

Let's have a look at the course's solution, and see if we can code review it.

Things I can spot are:

1. The `defaultBoard` is being recreated on every single render.
2. `firstUser` shouldn't be named `firstUser`.
3. The board goes from being an array of arrays to being a dictionary of arrays.
4. `newBoard[x][y]` still updates the existing array in-place.
5. `border="1"` is deprecated.
6. It might be easier to create the table in a loop.