Bringing Everything Together: Handling Button Clicks

The implementation of a button dispatching an onClick action

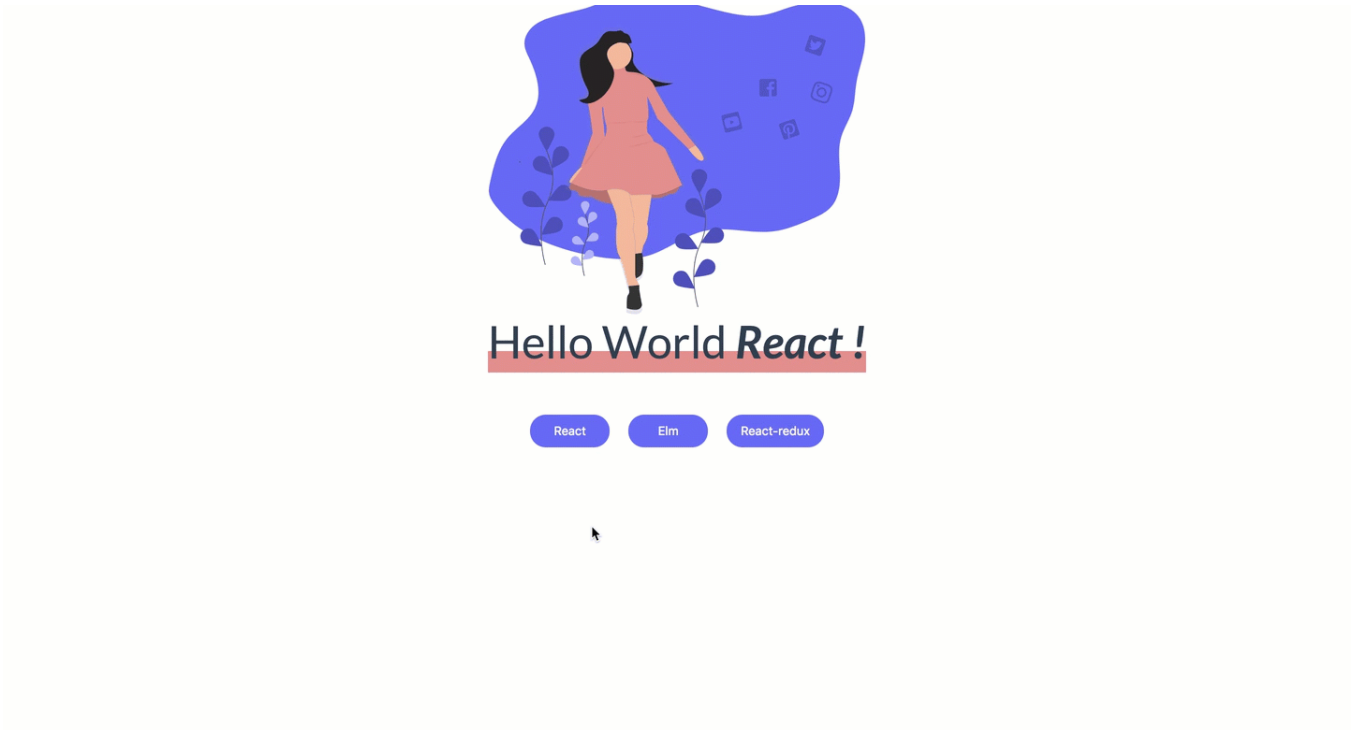
In the last lesson, we covered all but one aspect of the ButtonGroup implementation: the **data-tech attribute**. This is called a data attribute, which exists for each button.

It is a way to store some extra information that doesn't have any visual representation. It makes it slightly easier to grab certain values off of an element.

A completely rendered button will look like this:

```
<button
  data-tech="React"
  key="btn-1"
  className="hello-btn"> React
</button>
```

Right now, everything renders correctly, but upon clicking the button, nothing happens yet.



Well, that's because we haven't provided any click handlers yet. Let's do that now. Within the render function, let's set up an **onClick** handler:

```
<div>
  {technologies.map((tech, i) => (
    <button
      data-tech={tech}
      key={`btn-${i}`}
      className="hello-btn"
      onClick={dispatchBtnAction}
      >
      {tech}
    </button>
  ))}
</div>
```

Good. Let's write the **dispatchBtnAction** now.

Don't forget that the sole aim of this handler is to dispatch an action when a click has happened.

For example, if you click the react button, dispatch the action:

```
{
  type: "SET_TECHNOLOGY",
    tech: "React"
}
```

If you click the React-Redux button, dispatch this action:

```
{
  type: "SET_TECHNOLOGY",
    tech: "React-redux"
}
```

So, here's the **dispatchBtnAction** function.

```
function dispatchBtnAction(e) {
  const tech = e.target.dataset.tech;
  store.dispatch(setTechnology(tech));
}
```

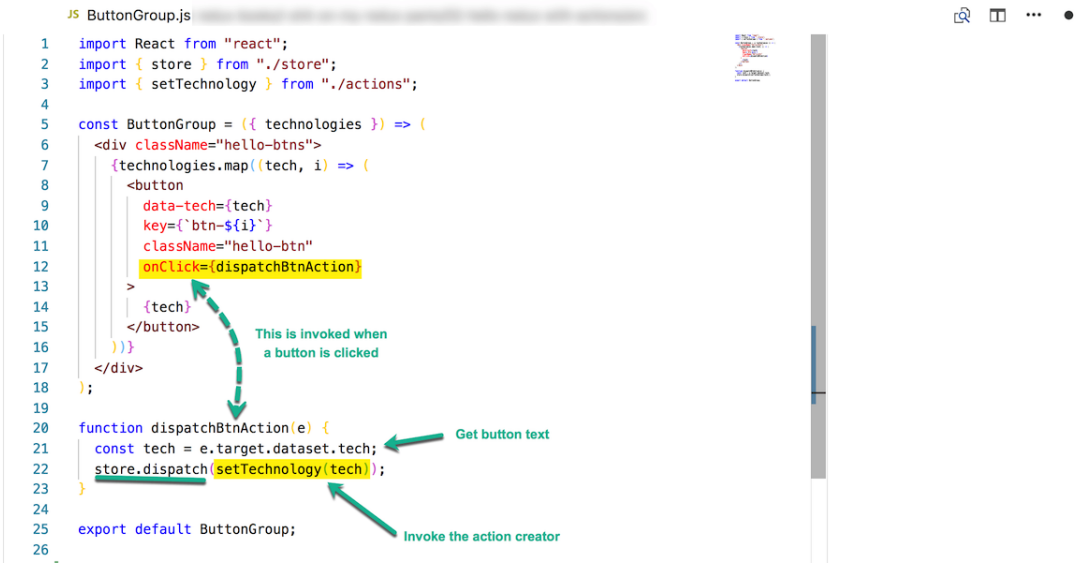
Hmmm. Does the code above make sense to you?

e.target.dataset.tech will get the data attribute set on the button, data-tech. Hence, tech will hold the value of the text.

store.dispatch() is how you dispatch an action in Redux, and setTechnology() is the action creator we wrote earlier!

```
function setTechnology (text) {
  return {
    type: "SET_TECHNOLOGY",
    text: text
  }
}
```

I have gone ahead to add a few comments in the illustration below, just so you understand the code.



Like you already know, **store.dispatch** expects an action object - nothing else. Don't forget the **setTechnology** action creator. It takes in the button text and returns the required action.

Also, the tech of the button is grabbed from the dataset of the button. You see, that's exactly why I had a data-tech attribute on each button. So we could easily grab the tech off each of the buttons.

Now we're dispatching the right actions. Can we ascertain that this works as expected now? We will find out in the next lesson.