Recap: Forms

Overview

Forms are an interesting topic in React, as they do not work like other DOM elements in React. This is because form elements keep their own state in regular HTML. Before moving on to your next exercise, it's important for you to understand this distinction between form elements and other elements in React, as well as how to work with them properly.

Form elements in React

Let's begin with an example of a form in HTML. Say you want to have a number input, so you write the following code:

```
<form>
    <label>
        Number of guests:
        <input type="number" name="guests" />
    </label>
    <button type="Submit">Submit</button>
</form>
```

The default behavior of an HTML form is to open a new page after the user clicks the submit button. In React, however, the usual practice is to use the **`preventDefault()`** method on an instance of the Event object that was built when the listened-to event was fired. Using the **`preventDefault()`** method stops the default behavior of the HTML form element. Once you've prevented the form's default behavior, you can use JavaScript and React to control the way that the form will be submitted. Additionally, after the **`preventDefault()`** is run, you can do whatever you need to do with the form data in a given scenario. There are two ways of achieving this: controlled and uncontrolled components.

Controlled and uncontrolled components

With controlled components, you let React take control of a given form's state. Although this requires some extra setup in your React code, the end result is that you can pass the form state value around in your React app, just like you can any other state value.

Alternatively, you can let the DOM control the state of your components, in which case you'd be using uncontrolled components. With uncontrolled components, you don't need to use event handlers for state updates in your form's elements. Instead, you can use **`refs`** to obtain the needed form element's values straight from the DOM.

The advantage of uncontrolled components is that sometimes it helps integrate React with other Front-end libraries. However, keep in mind that controlled components are the approach the official React docs suggest using most of the time.

Conclusion

Now that you've been re-acquainted with the concepts of forms in React, including controlled and uncontrolled components, you are better equipped to define the New Booking page in your app project in the next exercise. For a more in-depth recap of forms in React, you can revisit the following lesson items in the **Advanced React** course:

- What are controlled components?