

Exercise: Defining the Bookings page

Overview

Now that you've set up the foundations of the project, the next step is to begin adding the key features of the Little Lemon web app. In this exercise, you'll set up the table booking page.

Scenario

Currently, a visitor to the Little Lemon web app cannot reserve a table. Your task is to improve the app by allowing the user to enter data into the form so that they can complete their registration. Thus, you need to build a **Booking Form** component that has the following entries (form items):

- Date
- Time
- Number of guests
- Occasion (Birthday, Anniversary)
- Submit reservation button (to submit the form)

To do this, you need to implement a form in a controlled component named **BookingForm**. As you progress through the exercise, it may be helpful to revisit the following lesson items:

- [Creating a form](#)
- [What are controlled components?](#)
- [Creating a Form component in React](#)
- [Event handling and embedded expressions](#)
- [Using hooks](#)
- [Observing state](#)

Instructions

Step 1: Check the component and routes

If you haven't done so already, create the **BookingForm** and **BookingPage** components. The **BookingPage** will contain the **BookingForm** component, in addition to any additional content before and after the form.

Note: Before proceeding to the next step, check that your routes and navigation bar are set up to allow navigation to the booking page.

Step 2: Code the form structure

Next, you'll need to build the form structure in the **BookingForm** component. You can use the following plain HTML5 implementation as a starting point, however, you should convert it to JSX so that you can later connect the input to the

React state.

```
1 <form style="display: grid; max-width: 200px; gap: 20px">
2   <label for="res-date">Choose date</label>
3   <input type="date" id="res-date">
4   <label for="res-time">Choose time</label>
5   <select id="res-time ">
6     <option>17:00</option>
7     <option>18:00</option>
8     <option>19:00</option>
9     <option>20:00</option>
10    <option>21:00</option>
11    <option>22:00</option>
12  </select>
13  <label for="guests">Number of guests</label>
14  <input type="number" placeholder="1" min="1" max="10" id="guests">
15  <label for="occasion">Occasion</label>
16  <select id="occasion">
17    <option>Birthday</option>
18    <option>Anniversary</option>
19  </select>
20  <input type="submit" value="Make Your reservation">
21 </form>
```

Note: Keep in mind the difference between the `for` attribute in HTML and `htmlFor` in JSX. Also, remember to self-close all tags in JSX.

Step 3: Code the form behavior

Using what you already know about `events`, `effects` and `state` in React, update your form's code to keep track of its own state.

- Define a state variable for field in the form.
- Connect each `state` variable to the form fields using the `value` and `onChange` form element attributes.
- The options in the booking time field should be displayed from a list of available times. For now, create a stateful array in the component named `availableTimes` and use this state variable to populate the time select field options.

Tip: Use the `useState` function to declare the variable.

Now that the state is connected to the input elements, the form is ready to communicate with an API, a task you may complete in future.

Conclusion

By completing this exercise, you've created the component of the Bookings page for customers to reserve a table. In the next lesson, you'll work on lifting up the state and connecting the component to the list of available reservation times.