

Testing User Events Lab (CodeGrade)



[\(https://github.com/learn-co-curriculum/react-tdd-testing-user-events-lab\)](https://github.com/learn-co-curriculum/react-tdd-testing-user-events-lab) [\(https://github.com/learn-co-curriculum/react-tdd-testing-user-events-lab/issues/new\)](https://github.com/learn-co-curriculum/react-tdd-testing-user-events-lab/issues/new)

Learning Goals

- Use test-driven development to create a form and test form events

Introduction

In this lab, you will build on the portfolio site you created earlier in this section to get some practice at testing user events in React. You'll be responsible for writing tests as well as implementing the described functionality. You can check your work against the solution branch once you're finished.

Instructions

The requirement for this lab is to add a newsletter signup form to your portfolio page. The form should include:

- a text input for the user's name
- a text input for the user's email address
- a set of checkboxes allowing the user to select their interests
- a button to submit the form

When the form is submitted, use the form data to display a personalized message indicating that the form was submitted successfully.

Bonus: Include a list of the user's interests in the message as well.

You will be doing your coding in `src/App.js` and `src/__tests__/App.test.js`. Note that the `App.test.js` file includes the tests for the existing code for your reference. You will be responsible for creating the additional tests for the form and its behavior. We've provided descriptions of what kind of functionality to test, but you'll need write the actual tests.

The Process

Recall that, when testing user events, your tests should:

- verify that the initial state of the page is what you want
- simulate a user event (e.g., typing in a text box)
- verify that the state of the page updates as expected

Be sure to follow the test-driven development process in creating your tests and code. Specifically, for each feature of your app, you should:

- identify the desired element or behavior
- write the tests to check for that element or behavior
- write the code to make the tests pass

Remember also to use accessible queries as much as possible.

Resources

- [Testing Library: Queries ↗ \(https://testing-library.com/docs/queries/about\)](https://testing-library.com/docs/queries/about)
- [Jest DOM - Custom Matchers ↗ \(https://github.com/testing-library/jest-dom\)](https://github.com/testing-library/jest-dom)
- [Testing Library: user-event ↗ \(https://testing-library.com/docs/ecosystem-user-event/\)](https://testing-library.com/docs/ecosystem-user-event/)
- [MDN: ARIA Role Reference ↗ \(https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques\)](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques)

This tool needs to be loaded in a new browser window

Load Testing User Events Lab (CodeGrade) in a new window

