# Running Tests in React

New Attempt

**Due** No Due Date **Points** 1 **Submitting** a website url

## Learning Goals

- Use Jest to run tests on specific files

## Introduction

In this lab, we'll discuss how the tests are set up for the labs in this phase, and give some tips for running tests.

## Running Jest Tests

There are a number of JavaScript test frameworks out there. For testing vanilla JavaScript applications, one popular choice is Mocha, which we use for the tests in our JavaScript labs.

**Jest** **(https://jestjs.io/)** is another popular choice for JavaScript developers, and in particular the React community. Jest, like React, was developed by Facebook and is an open source project. You can read the **jest docs** **(https://jestjs.io/)** if you're curious to learn more.

Jest comes preinstalled when you generate a React project using `create-react-app`, so all you have to do to run tests in React labs is run `npm test`, which will execute the test script found in the `package.json` file. You can also run `learn test`, which will run the test files as well as sync your progress with Canvas.

Running `learn test` or `npm test` should produce output like this in your terminal:

This command will run all tests in the `src` directory by looking for files that have `.test.js` in the file name (you'll typically find them in the `__tests__` directory for our labs).

After running `learn test`, any changes you make to your components will cause the tests to run again, so you can keep the tests running as you work!

## Code Along

There are a couple of tests defined for this lab so you can get some practice.

To get started, run `npm install` (if you haven't already), then run `npm test` or `learn test`.

You should see something like this in the output:

```
 FAIL  src/__tests__/Header.test.js
  ● displays the text 'hello from the Header!'

    expect(received).toBeInTheDocument()

    received value must be an HTMLElement or an SVGElement.
    Received has value: null

       8 |   render(<Header />);
       9 |
    > 10 |   expect(screen.queryByText("hello from the Header!")).toBeInTheDocumen
         |                                                        ^
      11 | });
      12 |

      at __EXTERNAL_MATCHER_TRAP__ (node_modules/expect/build/index.js:342:30)
      at Object.<anonymous> (src/__tests__/Header.test.js:10:56)

 FAIL  src/__tests__/Article.test.js
  ● displays the text 'please pass this test'

    expect(received).toBeInTheDocument()

    received value must be an HTMLElement or an SVGElement.
    Received has value: null

       8 |   render(<Article />);
       9 |
    > 10 |   expect(screen.queryByText("please pass this test")).toBeInTheDocument
         |                                                       ^
      11 | });
      12 |

      at __EXTERNAL_MATCHER_TRAP__ (node_modules/expect/build/index.js:342:30)
      at Object.<anonymous> (src/__tests__/Article.test.js:10:56)

Test Suites: 2 failed, 2 total
Tests:       2 failed, 2 total
Snapshots:   0 total
Time:        3.486 s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
```

```
  › Press o to only run tests related to changed files.
  › Press q to quit watch mode.
  › Press p to filter by a filename regex pattern.
  › Press t to filter by a test name regex pattern.
  › Press Enter to trigger a test run.
```

Let's focus on the `Header.test.js` file first. To tell Jest to only run tests on this one file, press the **p** key in your terminal (this will let you filter out tests by their filename). In the next screen, type in `Header`:

```
 Pattern Mode Usage
  › Press Esc to exit pattern mode.
  › Press Enter to filter by a filenames regex pattern.

  pattern › Header

  Pattern matches 1 file
  › src/__tests__/Header.test.js
```

Then, press the Enter key to run tests in the `Header.test.js` file only.

See if you can get this test passing by updating the code in `/src/components/Header.js`!

Next, press the **a** key in your terminal to tell Jest to run **all** tests. Try getting the tests for the `Article` component to pass too.

When you've finished, you can hit the **q** key to exit Jest.

# Resources

- **Jest (https://jestjs.io/)**