Testing React Components

The recommended testing framework is Jest. This guide assumes that you followed the Unit testing guide to set up Jest.

The @testing-library/react by Kent C. Dodds has risen in popularity since its release and is a great replacement for enzyme. You can write unit and integration tests and it encourages you to query the DOM in the same way the user would. Hence the guiding principle:

> The more your tests resemble the way your software is used, the more confidence they can give you.

It provides light utility functions on top of `react-dom` and `react-dom/test-utils` and gives you the confidence that refactors of your component in regards to the implementation (but not functionality) don't break your tests.

Installation

Install the library as one of your project's `devDependencies`. Optionally you may install `jest-dom` to use its custom jest matchers.

```
npm install --save-dev @testing-library/react @testing-library/jest-dom
```

Create the file `setup-test-env.js` at the root of your project. Insert this code into it:

```
import "@testing-library/jest-dom/extend-expect"
```

This file gets run automatically by Jest before every test and therefore you don't need to add the imports to every single test file.

Lastly you need to tell Jest where to find this file. Open your `jest.config.js` and add this entry to the bottom after 'setupFiles':

```
module.exports = {
  testEnvironment: `jsdom`,
  setupFilesAfterEnv: ["<rootDir>/setup-test-env.js"],
}
```

> **Note:** The Jest 27 configuration `testEnvironment` default value changed to `node`, you need to change it to `jsdom` for React Testing Library to work properly

Usage

Let's create a little example test using the newly added library. If you haven't done so already, read the unit testing guide — essentially you'll use `@testing-library/react` instead of `react-test-renderer` now. There are a lot of options when it comes to selectors, this example chooses `getByTestId` here. It also utilizes `toHaveTextContent` from `jest-dom`:

```javascript
import React from "react"
import { render } from "@testing-library/react"

// You have to write data-testid
const Title = () => <h1 data-testid="hero-title">Gatsby is awesome!</h1>

test("Displays the correct title", () => {
  const { getByTestId } = render(<Title />)
  // Assertion
  expect(getByTestId("hero-title")).toHaveTextContent("Gatsby is awesome!")
  // --> Test will pass
})
```