

Intro to Testing with React Testing Library

By Chris Schmitz

Agenda

- What and why
- Types of tests
- Tools for writing tests
- Examples

What is a test?

Code you write to verify the behavior of your application.

Why write tests?

Documentation

Tests are a specifications for how our code should work.

Consistency

Verify that engineers are following best practices and conventions for your team.

Comfort & Confidence

A strong test suite is like a warm blanket.

Productivity

We write tests because it allows us to ship quality code faster.

Types of Tests

- **End-to-End:** Spin up your app and simulate user behavior. Kind of like a robot performing a task in your app.
- **Integration:** Verify that multiple units work together.
- **Unit:** Verify the functionality of a single function/component.
- **Static:** Catch typos and errors while writing code.

React Component Tests

A Basic Test

```
const expected = true;
const actual = false;

if (actual !== expected) {
  throw new Error(`${actual} is not ${expected}`);
}
```

Unhelpful Output

```
node tests/basic.test.js
/Users/chriisschmitz/code/presentation-react-testing/tests/basic.test.js:5
throw new Error(`${actual} is not ${expected}`);
^
```

Error: **true** is not **false**

```
at Object.<anonymous> (/Users/chriisschmitz/code/presentation-react-testing/tests/basic.test.js:5:5)
at Module._compile (internal/modules/cjs/loader.js:776:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
at Module.load (internal/modules/cjs/loader.js:653:32)
at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
at Function.Module._load (internal/modules/cjs/loader.js:585:3)
at Function.Module.runMain (internal/modules/cjs/loader.js:829:12)
at startup (internal/bootstrap/node.js:283:19)
at bootstrapNodeJSCore (internal/bootstrap/node.js:622:3)
```

A Jest Test

```
const expected = true;
const actual = false;

test("it works", () => {
  expect(actual).toBe(expected);
});
```

Jest Output

```
→ presentation-react-testing yarn jest tests/basic-jest.test.js
yarn run v1.19.0
$ /Users/chrisschmitz/code/presentation-react-testing/node_modules/.bin/jest tests/basic-jest.test.js
FAIL tests/basic-jest.test.js
  ✕ it works (6ms)

● it works

  expect(received).toBe(expected) // Object.is equality

  Expected: false
  Received: true

   3 |
   4 | test("it works", () => {
>  5 |   expect(actual).toBe(expected);
     |                  ^
   6 | });
   7 |

      at Object.<anonymous>.test (tests/basic-jest.test.js:5:18)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:   0 total
Time:        1.051s
Ran all test suites matching /tests\/basic-jest.test.js/i.
```

Questions?

Otherwise, let's write some tests!

More Things to Explore

- Clearer assertions with [jest-dom](#).
- Accessibility testing with [jest-axe](#).
- Testing hooks with [react-hooks-testing-library](#).
- More realistic event simulation with [user-event](#).
- The [code sandboxes from this talk](#).

Thanks!