Better testing with Enzyme

Dillon Mulroy | Software Engineer @ Formidable

Upfront 🔥 takes & key takeaways

- Starting a new project? Use React Testing Library
- Shallow rendering is bad, focuses on implementation details, and creates fragile tests
- Integration tests are far more important and useful than unit tests

Why React Test Library? 🐙

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

Why React Test Library? 🐙

The problem

You want to write maintainable tests for your React components. As a part of this goal, you want your tests to avoid including implementation details of your components and rather focus on making your tests give you the confidence for which they are intended. As part of this, you want your testbase to be maintainable in the long run so refactors of your components (changes to implementation but not functionality) don't break your tests and slow you and your team down.

Source: https://testing-library.com/docs/react-testing-library/intro

Why React Test Library? 🐙

The solution(s)

- Test and assert against actual rendered DOM nodes rather than instances of rendered React elements
- Force tests to run in such a way that mimics the way a user would actually use our applications/components
- Provide a small API footprint and simple utilities for finding and interacting with DOM elements to avoid testing implementation details.

Back to Enzyme

Shallow Rendering

Don’t do it.

Seriously.

Don’t.

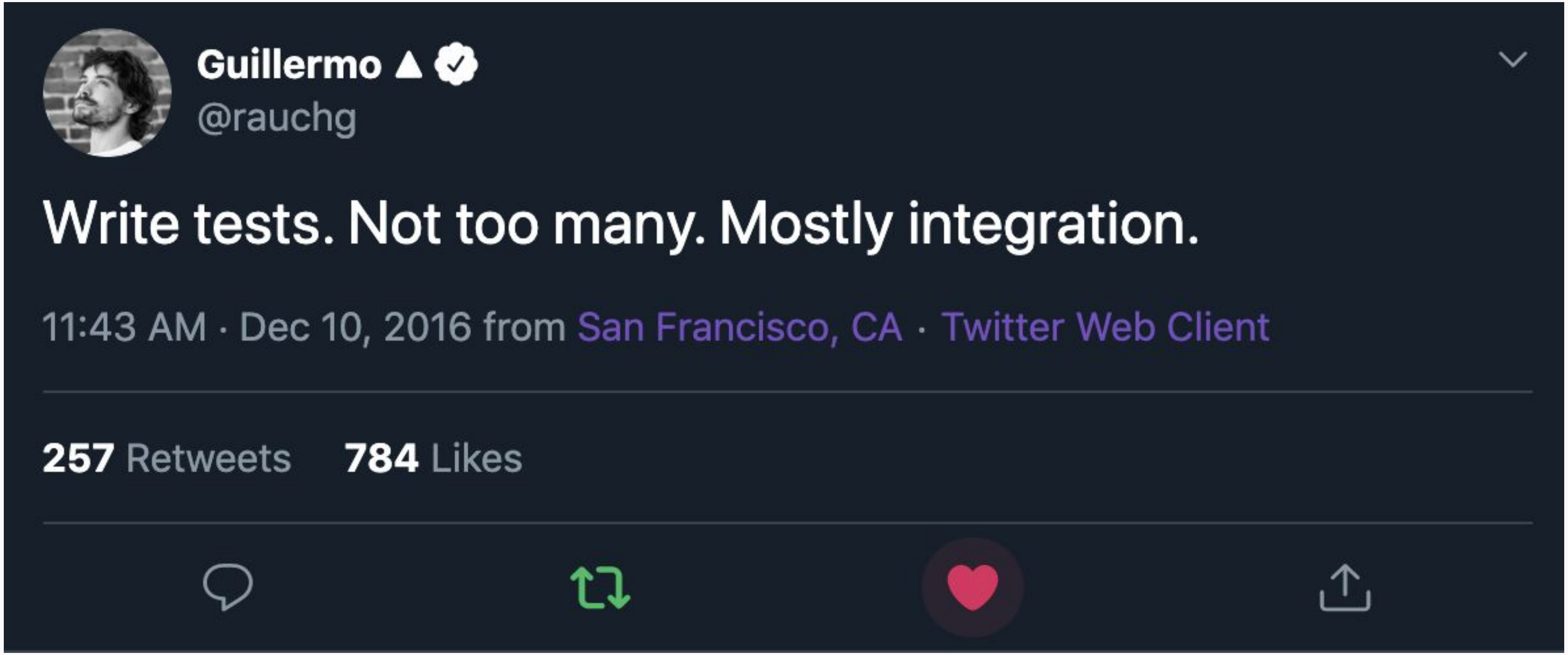
Shallow Rendering, don’t do it.

“With shallow rendering, I can refactor my component's implementation and my tests break. With shallow rendering, I can break my application and my tests say everything's still working.” - Kent C. Dodds

Demo

https://github.com/ReactMD/better-testing-with-enzyme

Integration Testing



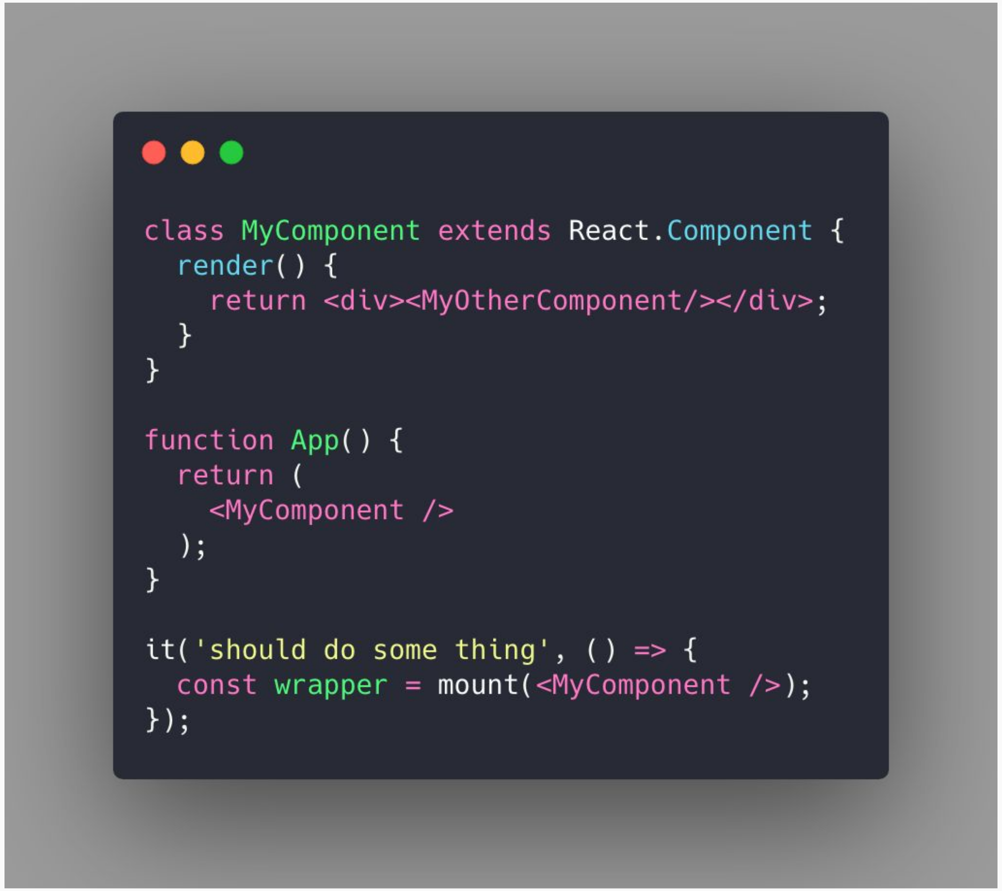
Guillermo ▲ ✓
@rauchg

Write tests. Not too many. Mostly integration.

11:43 AM · Dec 10, 2016 from San Francisco, CA · Twitter Web Client

257 Retweets **784** Likes

Unit Testing

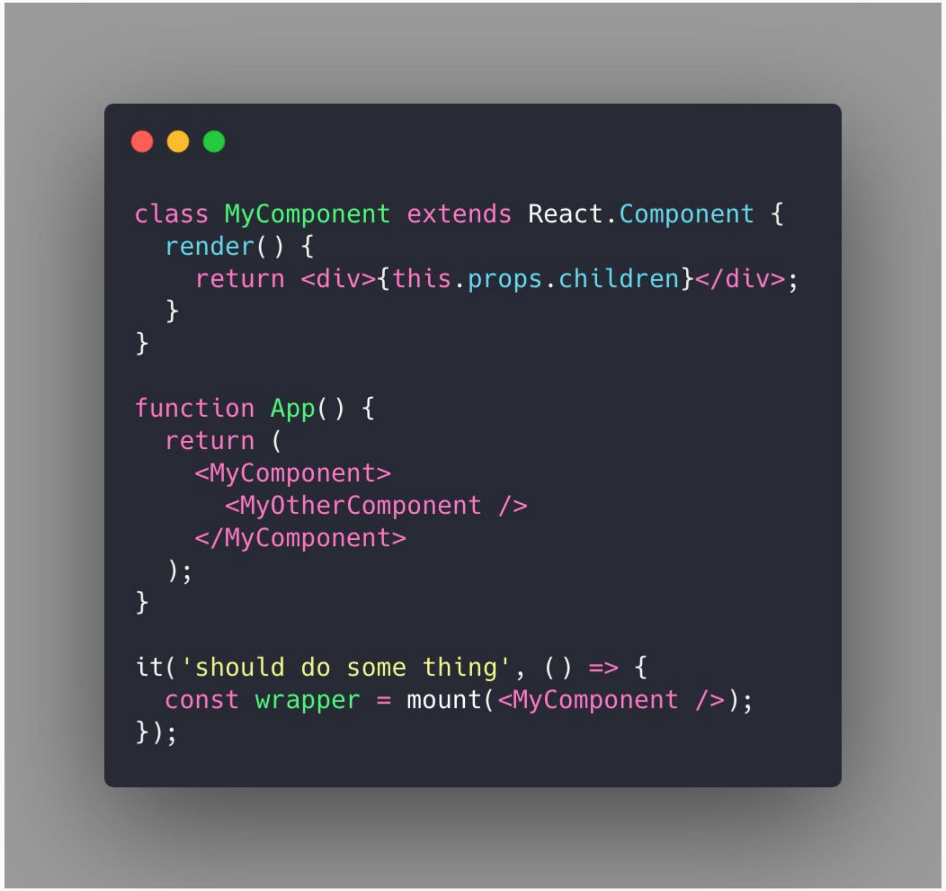


```
class MyComponent extends React.Component {
  render() {
    return <div><MyOtherComponent/></div>;
  }
}

function App() {
  return (
    <MyComponent />
  );
}

it('should do some thing', () => {
  const wrapper = mount(<MyComponent />);
});
```

Unit Testing



```
class MyComponent extends React.Component {
  render() {
    return <div>{this.props.children}</div>;
  }
}

function App() {
  return (
    <MyComponent>
      <MyOtherComponent />
    </MyComponent>
  );
}

it('should do some thing', () => {
  const wrapper = mount(<MyComponent />);
});
```

Resources

- https://testing-library.com/docs/react-testing-library/intro
- https://kentcdodds.com/blog/why-i-never-use-shallow-rendering
- https://kentcdodds.com/blog/testing-implementation-details

Find Me @

- github: https://github.com/dmmulroy
- twitter: https://twitter.com/dillon_mulroy
- gmail: dillon.mulroy@gmail.com