

How I Learned to Stop Worrying and Love the ~~Bomb~~ Component Testing



- Satvik Sharma
- Developer at Easy Agile
- Occasional flower crown wearer

What are we testing?

- Custom date picker field built using:
 - Atlaskit components
 - React + Hooks
- Supports setting min/max dates

Why?

- Existing buggy implementation
- Can't use Atlaskit's



How?

- Typescript
- Storybook
- React hooks
- @testing-library/jest-dom
- @testing-library/react
 - The library formerly known as react-testing-library

How?

- Typescript
- Storybook
- React hooks
- @testing-library/jest-dom
- @testing-library/react
 - The library formerly known as react-testing-library

What is component testing?

- Testing UI Components
- Snapshot testing
- Visual regression
- **Asserting against the DOM**

@testing-library

- Not specific to a test runner
- Wrappers for:
 - React
 - Angular
 - Vue
 - And more!

Approach

The more your tests resemble the way your software is used, the more confidence they can give you. — *Kent C. Dodds*

```
const HelloWorldSFC = (props) => <p>Hello world!</p>;
```

```
const HelloWorldFC = (props) => {  
  const [displayValue] = useState("Hello world!");  
  return <p>{displayValue}</p>;  
}
```

```
class HelloWorldClass extends React.Component {  
  render() {  
    return <p>Hello world!</p>;  
  }  
}
```

```
const components = [  
  HelloWorldSFC,  
  HelloWorldFC,  
  HelloWorldClass  
];
```

```
describe("Hello world!", () => {  
  it("should render", () => {  
    for (const Component of components) {  
      const { getByText } = render(<Component />);  
      expect(getByText("Hello world!")).toBeInTheDocument();  
  
      // Clean up the DOM, this is usually done for us  
      // between tests automatically  
      cleanup()  
    }  
  });  
});
```





A-80



$$x^2 + 8x + 12 = 0$$

$$\frac{x^2 + 8x + 12}{x^2 + 8x + 12} = 1$$

$$4y^4y$$

$$2x10^{-2}$$

$$\frac{12x-1}{y^2/12}$$

$$\begin{matrix} 9 \\ \times \end{matrix}$$

「タイ
ワーク」

Counter



0



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

import React from "react";
import { fireEvent, render } from "@testing-library/react";

it("should render 0 initially", () => {
  const { getByText } = render(<Counter />);
  const counterValue = getByText("0");
  expect(counterValue).toBeInTheDocument();
});

it("should have a disabled subtract button when rendering 0", () => {
  const { getByText } = render(<Counter />);
  const subtractButton = getByText("-");
  expect(subtractButton).toBeDisabled();
});

it("should have a disabled add button when rendering 10", () => {
  const { getByText, getByTitle } = render(<Counter />);
  const addButton = getByText("+");

  for (let i = 0; i < 10; i++) {
    fireEvent.click(addButton);
  }

  expect(getByText("10")).toBeInTheDocument();
  expect(addButton).toBeDisabled();
});
```



```
import "@testing-library/jest-dom/extend-expect";
import React from "react";
import { fireEvent, render } from "@testing-library/react";
import Counter from "../Counter";

it("should render 0 - 10", () => {
  const { getByText, getByTitle } = render(<Counter />);
  const addButton = getByText("+");

  expect(getByText("0")).toBeInTheDocument();

  for (let i = 0; i < 10; i++) {
    fireEvent.click(addButton);
    expect(getByText(`${i + 1}`)).toBeInTheDocument()
  }
});
```

```
~/s/ea-kit (EA-1015-issue-status-colors|+ 2...) » yarn run test --coverage
```

```
yarn run v1.17.3
```

```
$ jest --coverage
```

```
PASS src/components/DatePicker/index.test.tsx (5.252s)
```

```
DatePicker
```

```
Open / close behaviour
```

- ✓ should have the calendar closed when not focused (77ms)
- ✓ should open the calendar if focused (149ms)
- ✓ should open the calendar if clicked on (64ms)
- ✓ should close the calendar if clicked away from (57ms)
- ✓ should keep the calendar open if a non close key is pressed (69ms)
- ✓ should close the calendar if a close key, i.e. Escape, is pressed while open (237ms)
- ✓ should close the calendar when a date is selected (51ms)

```
Input and calendar sync with value
```

- ✓ should have an input matching the selectedDate (7ms)
- ✓ should have the selected date selected in the calendar (58ms)
- ✓ should update the input when a date is selected (55ms)
- ✓ should update the selected calendar date when the input changes to a valid date (93ms)
- ✓ should update the selected calendar date when the input changes to a valid date when using external validation (82ms)
- ✓ should have the last valid input value selected in the calendar when given an invalid input (96ms)

```
Min / max date behaviour
```

- ✓ should disable dates before the min date (158ms)
- ✓ should disable dates after the max date (246ms)
- ✓ should update the disabled dates when the min date changes (98ms)
- ✓ should update the disabled dates when the max date changes (128ms)
- ✓ should ignore the min date if it's invalid (55ms)
- ✓ should ignore the max date if it's invalid (84ms)

```
Input validity based on rendered value
```

- ✓ should be valid if the initial value is empty and the field isn't required (6ms)
- ✓ should be invalid if the initial value is empty and the field is required (10ms)
- ✓ should be invalid if the initial value is not a date (8ms)
- ✓ should be invalid if the initial value is before the min date (16ms)
- ✓ should be invalid if the initial value is after the max date (30ms)
- ✓ should be invalid if the date is set outside of the min and max dates (98ms)
- ✓ should be invalid if the min and max dates change and the selected date is outside of the new range (28ms)
- ✓ should use the passed in isValid over the internal one (15ms)

```
Month change behaviour
```

- ✓ should not let you change the selected month if all rendered dates would be disabled (128ms)
- ✓ should use the month of the day after the min date if the initial value is before the minDate (43ms)
- ✓ should use the month of the day before the max date if the initial value is after the maxDate (45ms)
- ✓ should update the selected calendar month if the minDate is updated past the selected value (82ms)
- ✓ should update the selected calendar month if the maxDate is updated before the selected value (194ms)
- ✓ should not update the selected calendar month if the minDate is updated but the selected value is still in range (67ms)
- ✓ should update the selected calendar month if the maxDate is updated before the selected value (70ms)

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	98.06	94.44	100	97.58	
components/DatePicker	98.48	93.75	100	97.73	
index.tsx	98.48	93.75	100	97.73	40
helpers	100	100	100	100	
isDateInRange.ts	100	100	100	100	
hooks	97.59	93.75	100	97.3	
useCloseBehaviour.ts	100	93.75	100	100	3
useDateRange.ts	96.97	93.75	100	96.61	40,49

```
Test Suites: 1 passed, 1 total
```

```
Tests: 34 passed, 34 total
```

```
Snapshots: 0 total
```

```
Time: 6.825s
```

```
Ran all test suites.
```

```
🌟 Done in 8.32s.
```

Conclusions

- Still should only test components if needed
- Test runner needs to know JSX
- No garden paths

Questions