React – Testing with jest

In this lab, you will test a react component using jest.

# Objectives

In this lab, you will

* Test a Component from an existing project
* Test the react component

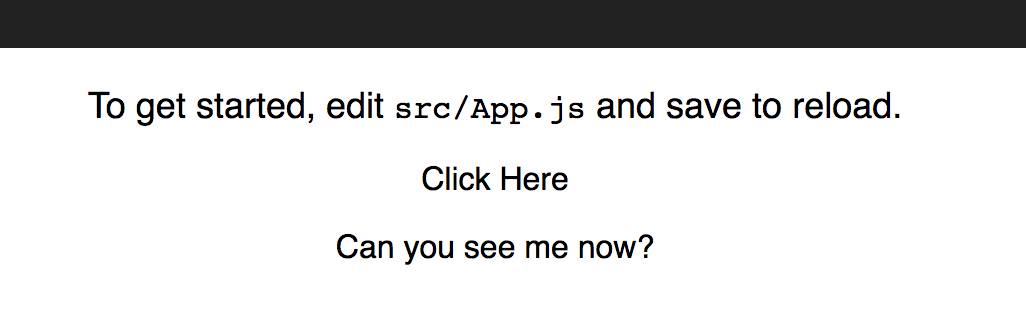
# Create the Project

1. Open the lab/testing folder and start the application with:

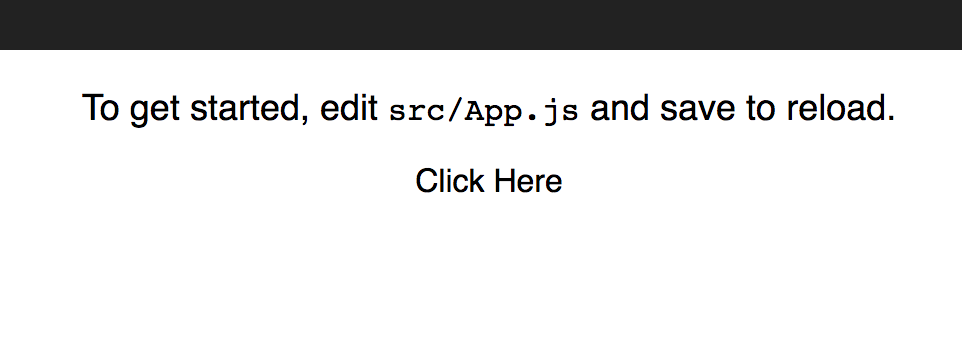
yarn install

yarn start

1. Open the browser to <http://localhost:3000/> and see the below:



1. Click where indicated and notice the line below it disappears!



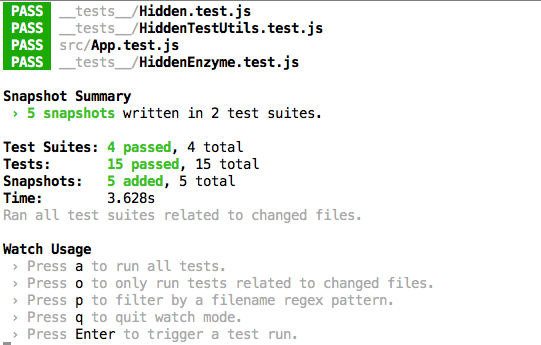
1. The <Hidden> component displays the messages and handles the click as shown in the presentation.

# Start the Tests

1. Open a terminal in the labs/testing folder, start the test with:

yarn test

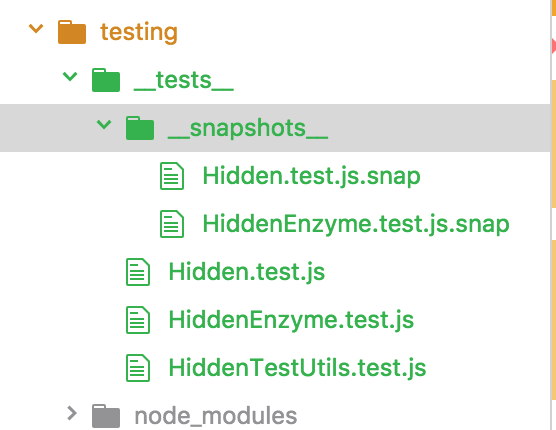
1. You should see the following:



1. All kinds of tests are succeeding because they are mostly comparing snapshots to previous ones that don’t exist yet. Type the ‘o’ command and jest will re-run the tests anytime a file changes.

# The \_\_tests\_\_ folder

1. Examine the \_\_tests\_\_ folder shown below:



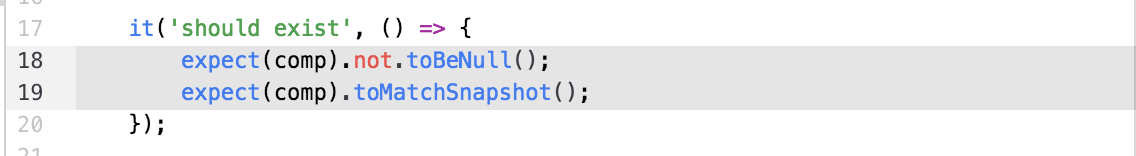
1. The \_\_snapshots\_\_ folder contains the snapshots (duh…)
2. The unit tests have the name \*.test.js. The three tests demonstrate different ways of writing tests. All use jest as the test runner.

# Using the react-test-renderer

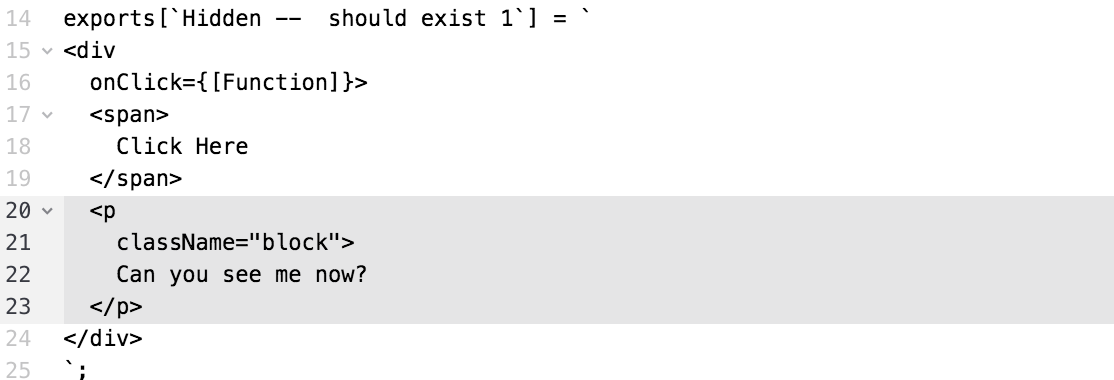
1. The test file, Hidden.test.js, demonstrates the usage of the react-test-renderer. The following steps analyze sections of the test program.



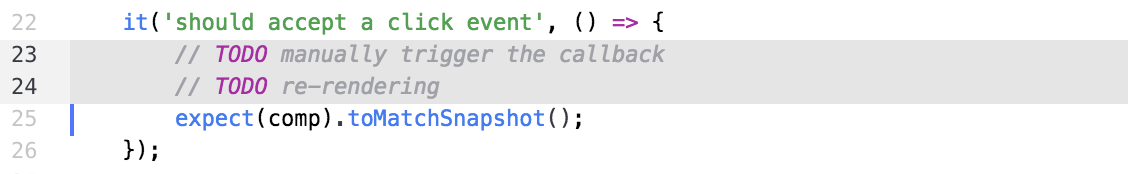
1. In the above, the component variable contains the rendered component, <Hidden>. The comp variable contains the JSON representation of the component. The following tests use these two variables extensively.



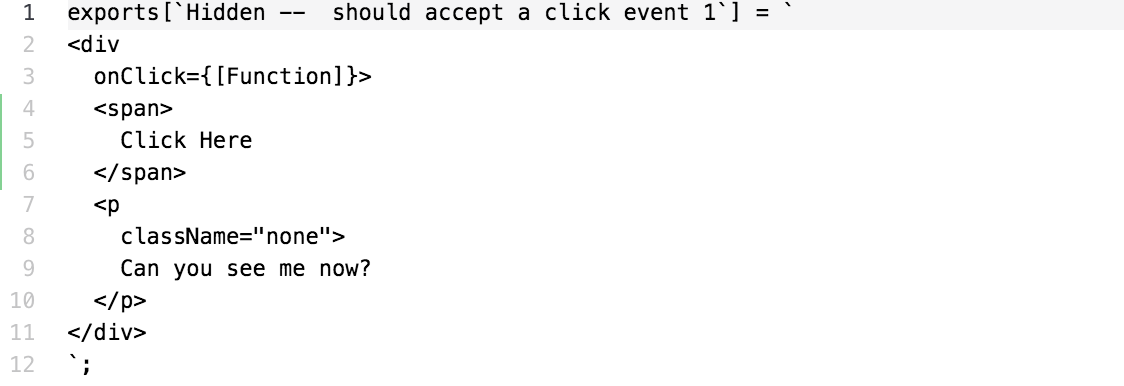
1. In the above, the first test verifies that the component rendered and takes a snapshot of it shown below:



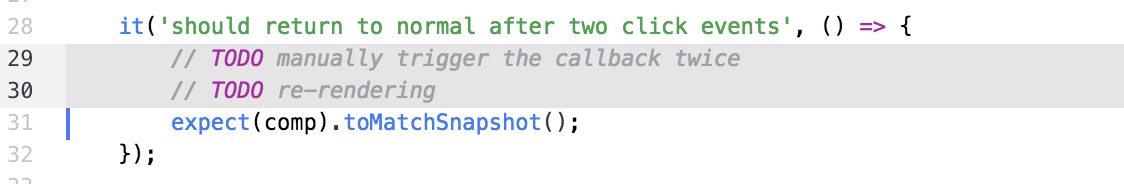
1. In the above snapshot, notice the className field on the <p> component is ‘block’ as expected.



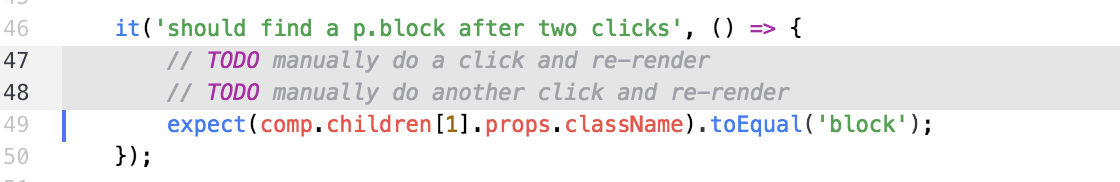
1. As shown above, the second test will manually trigger the callback by onClick method on the props hash. You can call the method with comp.props.onClick(). Convert the component to JSON again with component.toJSON() then take another snapshot as shown below:



1. In the above, notice the className attribute is now ‘none’ as expected.



1. In the above, trigger the callback twice and verify the className is “block” again via the snapshot.



1. In the last test above, trigger the callback twice and verify the className is “block” again by examining it programmatically.

# Extra Credit - Enzyme

1. Examine the Enzyme test program in \_\_tests\_\_/HiddenEnzyme.test.js.
2. Implement the two tests with TODO elements. Have fun!

Congratulations. You have completed this lab.