React Mock Testing

In this lab, you will test a react app using jest.fn() to define mock callbacks.

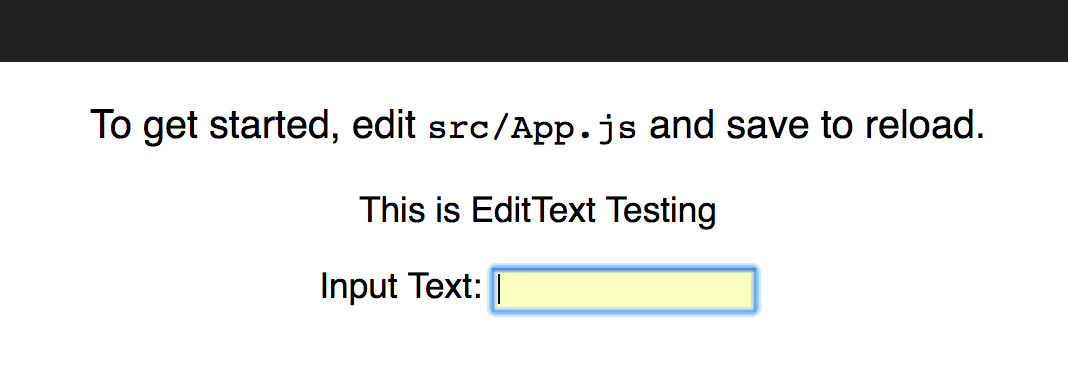
# Objectives

In this lab, you will

* Create unit tests using Enzyme
* Create and use a mock function for the EditText callback

# Examine the Application

1. Change to the lab/testing folder and install and start the app using yarn (or npm).
2. The app should open a browser window to <http://localhost:3000> and display the app as shown below:



1. Enter some text and notice the alert() message.
2. The EditText.js application looks like:

**import** React, { Component } from **'react'  
  
export default class** EditText **extends** Component {  
 constructor( props ) {  
 **super**(props)  
 **this**.state = {  
 text: **''** }  
 }  
 onSubmit = (e) => {  
 e.preventDefault()  
 **this**.props.endEdit( **this**.state.text )  
 **this**.setState({text: **''**})  
 }  
 onChange = (e) => {  
 **this**.setState({text: e.target.value })  
 }  
 render() {  
 **return** (  
 <**div**>  
 <**form onSubmit=**{**this**.onSubmit}>  
 <**label htmlFor="editText"**>Input Text: </**label**>  
 <**input  
 type="text"  
 onChange=**{ **this**.onChange }  
 **value=**{**this**.state.text}  
 **id="editText"** />  
 </**form**>  
 </**div**>  
 )  
 }  
}

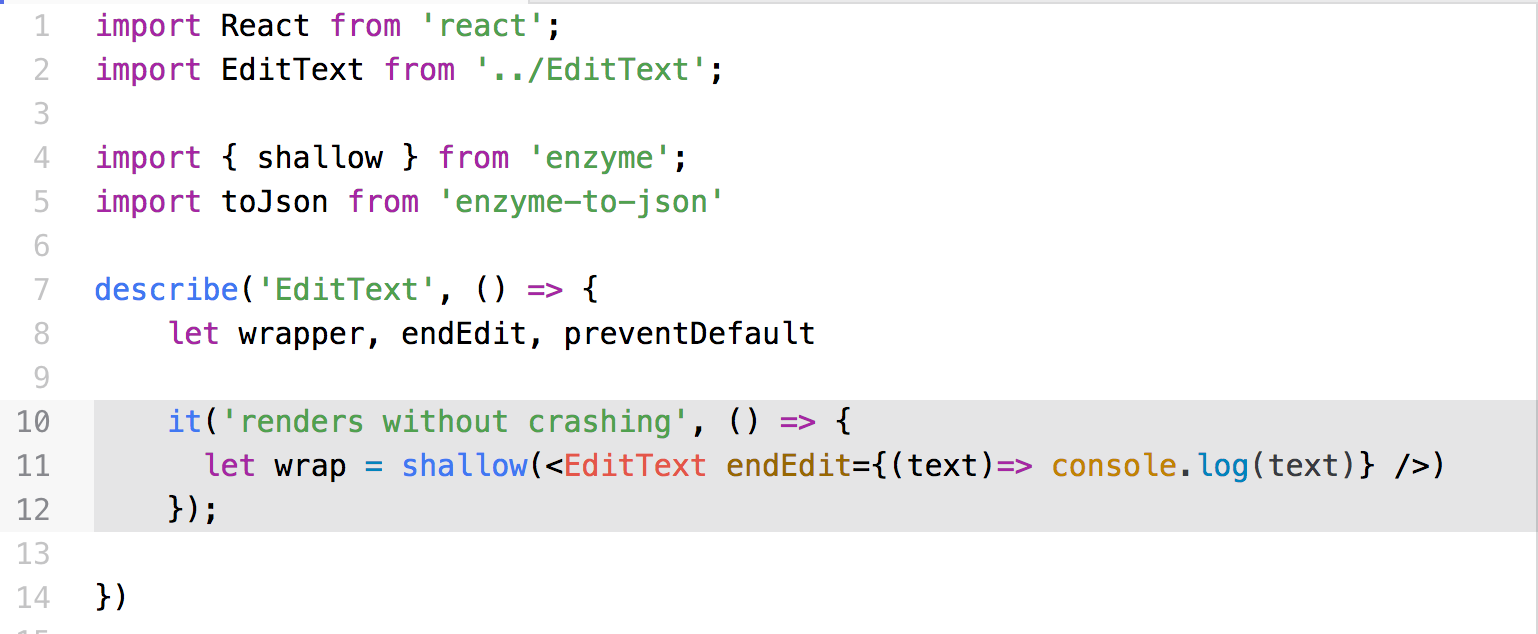
1. Notice the two callback, onSubmit() and onChange().
2. How many parameters do they have??
3. What type is the parameter??
4. Stop the application and start testing using

yarn test

1. Notice the test runs two tests.
2. What are the names of the test programs?

# Examine lab/testing/src/\_\_tests\_\_/EditText-test.js

1. Examine the test program shown below



1. Notice it contains only one test which verifies that enzyme can render it.
2. Notice line 8 defines global variables for all the test functions.

# Add the Setup Function

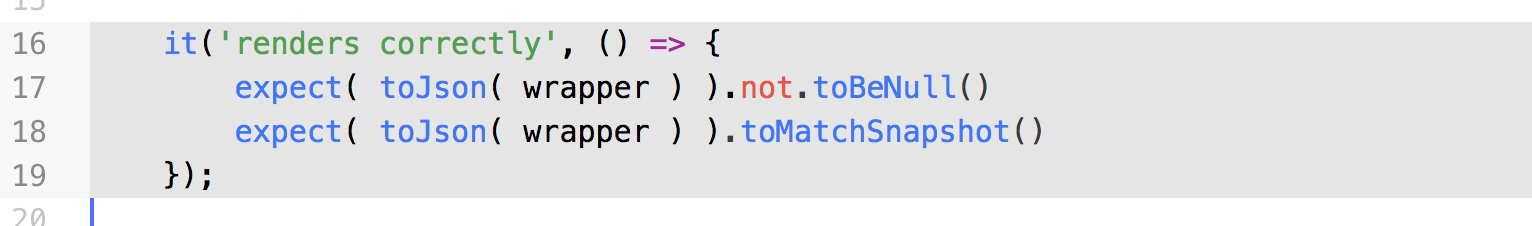
1. Insert the following setup function in the test program.



1. Notice this sets up the global variables used by the test functions.

# Take the First Snapshot

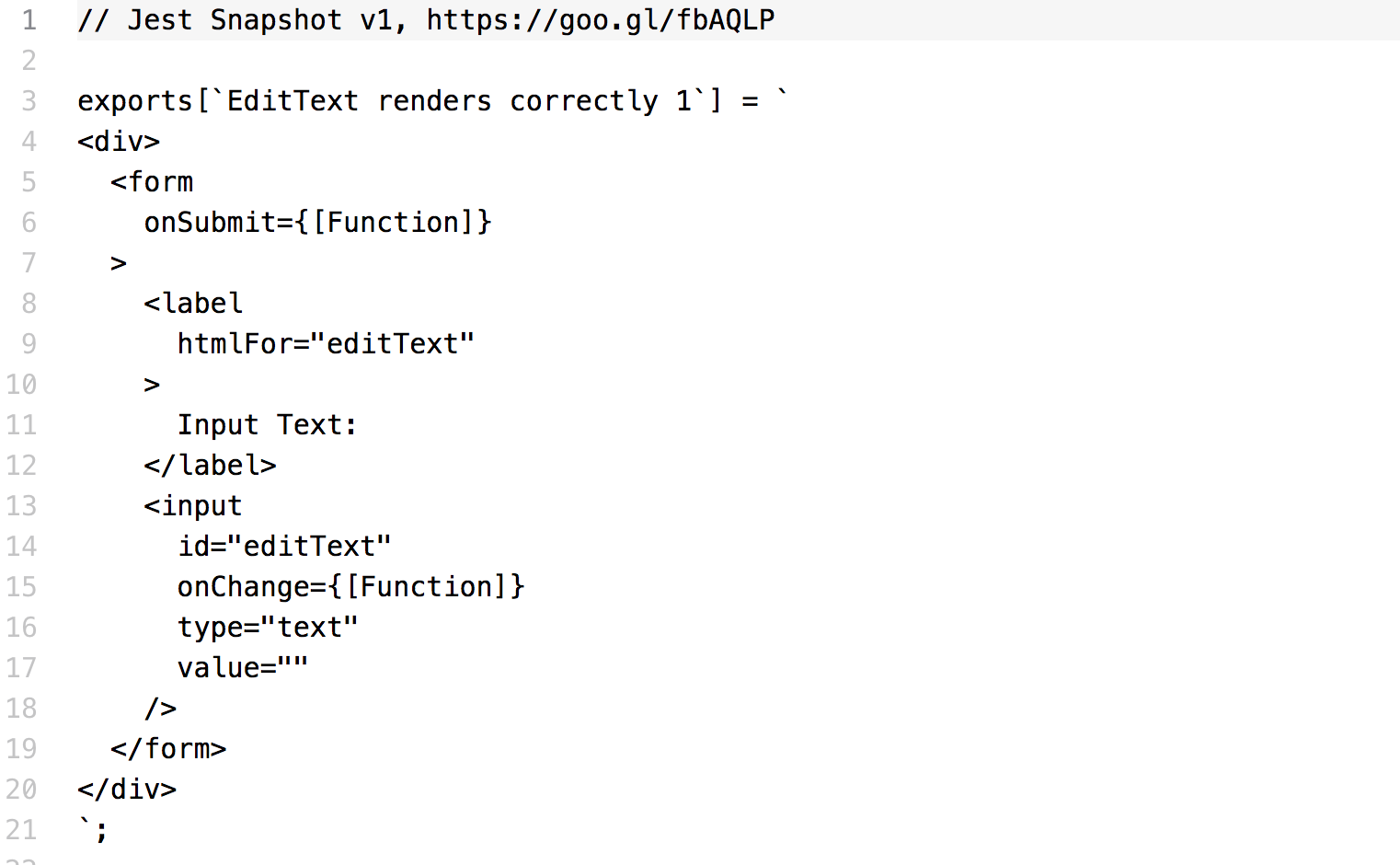
1. Add the following code to take the first snapshot.



1. Notice the call to toJson() converting the wrapper to a JSON object. Jest understands how to take a snapshot from a JSON object OR a string.
2. Where did the snapshot go?

# Examine the Snapshot

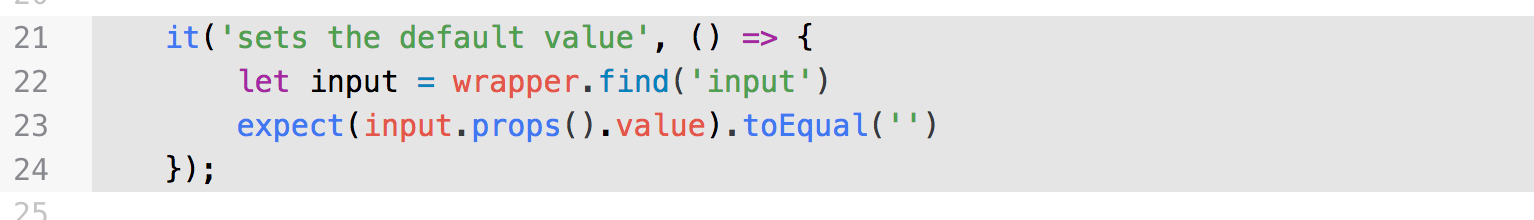
1. Open the file, lab/testing/src/\_\_tests\_\_/\_\_snapshots\_\_/EditText-test.js.snap as shown below:



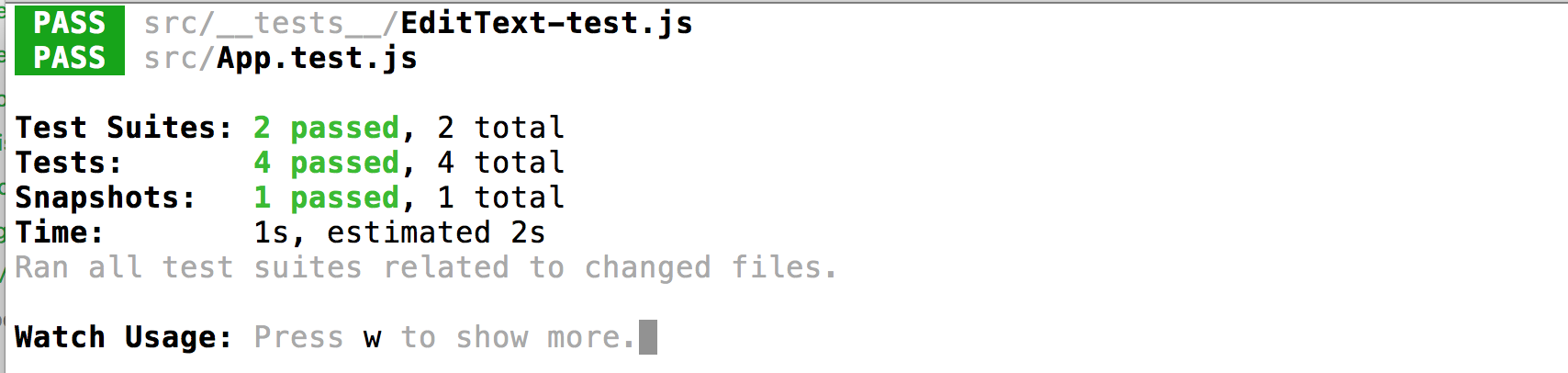
1. Enzyme renders the component to the above HTML.
2. Line 3 describes the text of the test function that takes the snapshot. The number 1 at the end of the line indicates the first snapshot in that function.

# Verify the Default Text Value

1. Add the following test to verify the default text value.



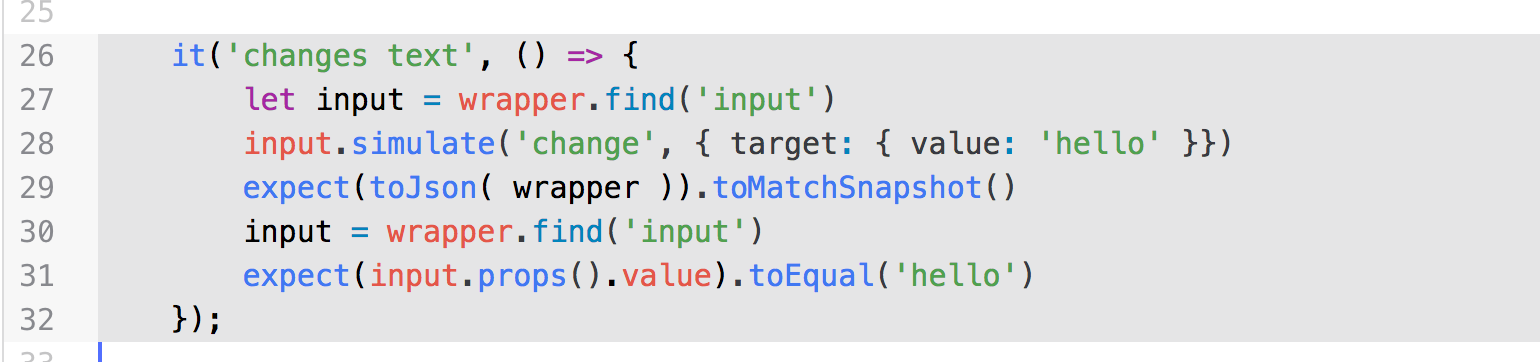
1. Notice the terminal window where we started jest using yarn test as shown below



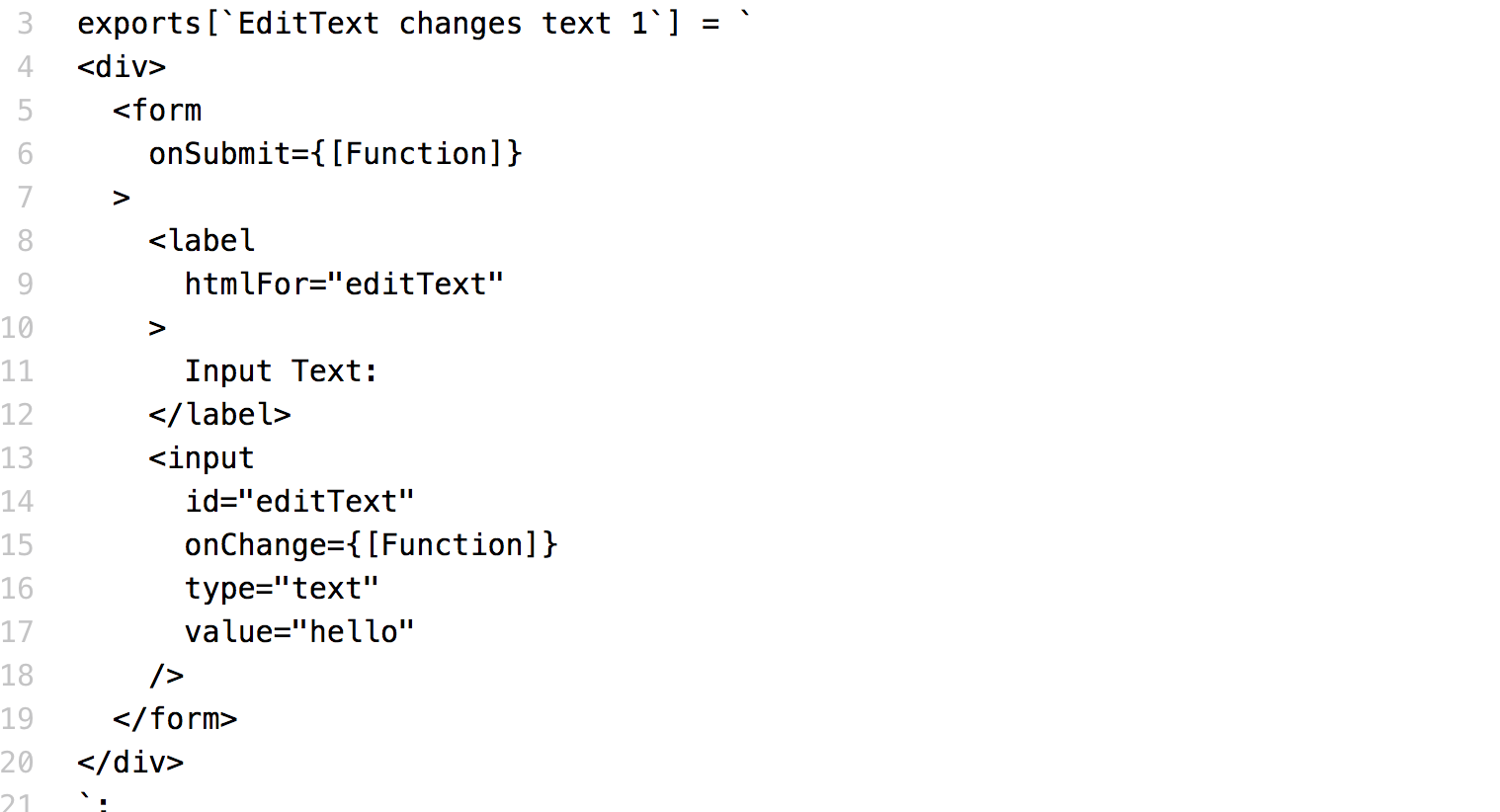
1. Each time we modify the test program, jest re-runs the tests.

# Change the Text

1. Let’s change the text, verify the change, and take another snapshot with the following code.



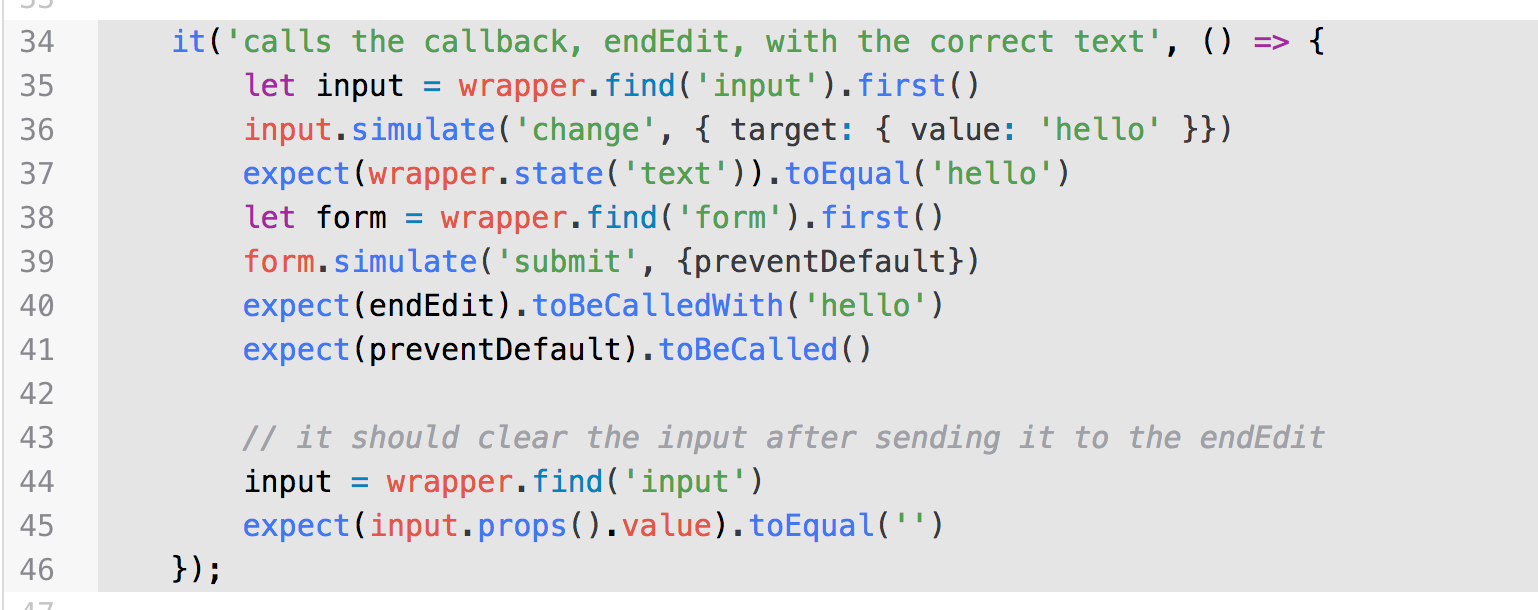
1. If you examine the snapshot again, the second snapshot shows the value on the input component to be “hello” just as we expect.



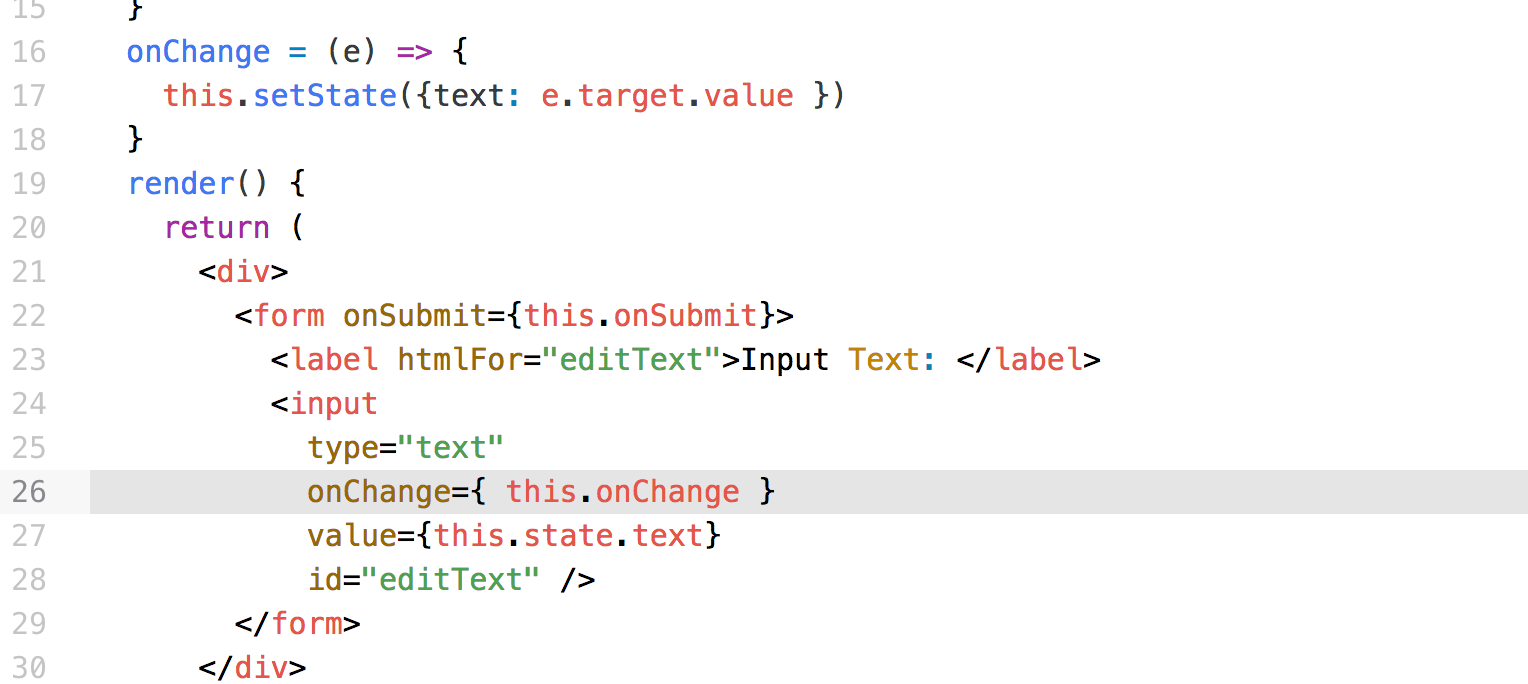
1. Notice line 17 shows the new value.

# Ensure the Test Calls the Callbacks

1. Verify the component calls the mock functions on the onChange and onSubmit events using the following test.



1. On line 36, how many parameters does the input.simulate() function have?
2. The first parameter tells jest to simulate the change event. Jest calls the onChange() function defined in the attribute in the component as shown below:



1. In the above, enzyme renders the component and configures the onChange() method to be connected to the onChange attribute in the <input> component.
2. How many parameters does the onChange() function have?
3. Enzymes calls the function in the component and uses the Event object we defined in the test case.
4. Lines 40 and 41 show how jest verifies the calling of the mock functions by the event handlers in the component.

Congratulations. You have completed this lab.