**Unit Testing React Form with Jest and Enzyme**

[Jest](https://jestjs.io/) is a JavaScript unit testing framework created by Facebook. I usually use mocha for unit testing Node.js code, but when it comes to testing React, Jest combined with Enzyme is pretty much the best. [Enzyme](https://github.com/airbnb/enzyme/issues/308) is a JavaScript utility library created by Airbnb. It lets you simulate DOM rendering and actions on the components and integrates well with Jest.

In this post, I am focusing on testing React form. When I first started writing unit tests on React form components, I spent a bit of time investigating and experimenting different ways. I especially struggled to write a unit test on multi-select and checkbox component. So, here it is.

**Focus**

* Testing a simple text input
* Testing checkbox
* Testing Multi-select options.
* Testing alert() function on form submit.

**Complete Solution**

This blog focuses purely on unit test code. You can check out the complete solutions here: [react-form-unit-test-example](https://github.com/mydatahack/react-form-unit-test-example).

**Unit Test Code**

[Here](https://github.com/mydatahack/react-form-unit-test-example/blob/master/src/components/Form.jsx) is the main Form component to be tested.

1. **Testing Text Input**

Testing text input is relatively simple. First mount the component with Enzyme’s mount function, then set the value and simulate the change. When there are multiple input fields, we can use .at(0) to find the input element by index.

<pre class=" toolbar:2 lang:js decode:true" >

</pre>

1. **Testing Checkbox**

For checkbox, assign the value for checked instead of value and then simulate change.

<pre class=" toolbar:2 lang:js decode:true" >

</pre>

1. **Testing Multi-Select**

To test multi-select, you need to set the individual option’s selected property to true. This solution works.

<pre class=" toolbar:2 lang:js decode:true" >

</pre>

Simulating change on the select element as below doesn’t work.

<pre class=" toolbar:2 lang:js decode:true" >

</pre>

1. **Testing alert() on Submit**

On submit, it will call the alert function with all the input values (see [here](https://github.com/mydatahack/react-form-unit-test-example/blob/master/src/components/Form.jsx)). To test this logic, we need to mock window.alert function and check if it is called with the correct argument, including all the input values. First, set the state and simulate submit. Then, check to see if mocked alert function is called with the value from the state.

<pre class=" toolbar:2 lang:js decode:true" >

</pre>