Component Unit Tests

For implementing unit tests in React applications, Mark Volkmann recommends using Jest and react-testing-library.

Snapshot Tests

Most components should have snapshot tests that assert what they render when given specific prop values.

Here is an example of implementing snapshot tests for the Hello component in src/components:

```
import React from 'react';
import {snapshot} from './test-util';

import Hello from './hello';

test('should render with no props', () => snapshot(<Hello />));

test('should render with name prop', () =>
  snapshot(<Hello name="Mastercard" />));
```

Other Tests

Components that support user interaction should also have tests that assert the state of the component or the state of the app after specific input events are triggered.

Here is an example of such a test for the LabeledSlider component in src/components that uses app state:

```
import React, {useContext} from 'react';
import {fireEvent, render} from '@testing-library/react';
import {Context, GrpContext, GrpProvider} from './grp-context';
import {getState} from './state-manager';

import LabeledSlider from './labeled-slider';

let context: Context;
const statePath = 'user.points';

// Using this wrapper around the component being tested
// provides a way to get the context object
// which is needed to use the getState function.
function Container() {
  // useContext can only be called in a function component.
  context = useContext(GrpContext);
  return (
    <LabeledSlider label="My Slider" min={10} max={90} statePath={statePath} />
  );
}

test('should work', () => {
  const {container} = render(
    <GrpProvider>
      <Container />
    </GrpProvider>
  );

  const input = container.querySelector('input');
  expect(input).not.toBeNull();

  if (input) {
    const initialValue = getState(context, statePath);
    expect(input.value).toBe(String(initialValue));

    const value = 25;
    fireEvent.change(input, {target: {value}});

    expect(input.value).toBe(String(value));
    expect(getState(context, statePath)).toBe(value);
  }
});
```