Testing Actions

We'll test Redux actions in our Weather app that's built with React and Redux.

Create a new file called `actions.test.js` in the `src/__tests__ /` folder. (create that if you haven't already) Let's start by testing the good ol' `changeLocation` action. Add the default structure, we'll need to import the action we want to test and `describe` "actions" and `changeLocation`:

```
// actions.test.js
import {
  changeLocation
} from '../actions';

describe('actions', function() {
  describe('changeLocation', function () {

  });
});
```

There's two things we want to verify of our action function: that it has the correct type and that it passes on the data we tell it to pass on. Let's verify the type of the `changeLocation` action is `'CHANGE_LOCATION'`:

```
// actions.test.js
/* … */
describe('changeLocation', function () {
  it('should have a type of "CHANGE_LOCATION"', function() {
    expect(changeLocation().type).toEqual('CHANGE_LOCATION');
  });
});
```

Run `npm run test` in the console and this is what you should see:

```
PASS  src/__tests__/actions.test.js (0.525s)
1 test passed (1 total in 1 test suite, run time 0.55s)
```

Nice, let’s verify that it passes on the location we pass into it:

```
// actions.test.js
/* … */
describe('changeLocation', function () {
  it('should have a type of "CHANGE_LOCATION"', function() {
    expect(changeLocation().type).toEqual('CHANGE_LOCATION');
  });

  it('should pass on the location we pass in', function() {
    var location = 'Vienna, Austria';
    expect(changeLocation(location).location).toEqual(location);
  });
});
```

Nice! Now let’s do the same thing for the `setSelectedDate` and `setSelectedTemp` action! First, `import` those two actions at the of the file and add the `describe` and `it`s:

```
describe('actions', function() {
  describe('changeLocation', function () { /* … */ });

  describe('setSelectedDate', function() {
    it('should have a type of SET_SELECTED_DATE', function() { });

        it('should pass on the date we pass in', function() { });
  });

  describe('setSelectedTemp', function() {
    it('should have a type of SET_SELECTED_TEMP', function() { });

        it('should pass on the temp we pass in', function() { });
  });
});
```

First let’s verify our `setSelectedDate` works as expected:

```
describe('actions', function() {
  describe('changeLocation', function () { /* … */ });

  describe('setSelectedDate', function() {
    it('should have a type of SET_SELECTED_DATE', function() {
            expect(setSelectedDate().type).toEqual('SET_SELECTED_DATE');
        });
        it('should pass on the date we pass in', function() {

            var date = '2016-01-01';
            expect(setSelectedDate(date).date).toEqual(date);
        });
  });

  describe('setSelectedTemp', function() {
    it('should have a type of SET_SELECTED_TEMP', function() { });

        it('should pass on the temp we pass in', function() { });
  });
});
```

and then our `setSelectedTemp`:

```
describe('actions', function() {
  describe('changeLocation', function () { /* … */ });

  describe('setSelectedDate', function() {
    it('should have a type of SET_SELECTED_DATE', function() {
            expect(setSelectedDate().type).toEqual('SET_SELECTED_DATE');
        });

        it('should pass on the date we pass in', function() {
            var date = '2016-01-01';
            expect(setSelectedDate(date).date).toEqual(date);
        });
  });

  describe('setSelectedTemp', function() {
    it('should have a type of SET_SELECTED_TEMP', function() {
            expect(setSelectedTemp().type).toEqual('SET_SELECTED_TEMP');
        });

        it('should pass on the temp we pass in', function() {
            var temp = '31';
            expect(setSelectedTemp(temp).temp).toEqual(temp);
        });
  });
});
```

Not too hard, huh? Run `npm run test` in your console now, and this is what you should see:

```
PASS  src/__tests__/actions.test.js (0.531s)
6 tests passed (6 total in 1 test suite, run time 0.554s)
```

Now go on and test the other actions too, I'll be here waiting for you! (skip the `fetchData` action, one negative aspect of thunks is how hard they are to test so we'll skip it)

Back? Everything tested? You should now see something like this in your console when running `npm run test`:

```
PASS  src/__tests__/actions.test.js (0.357s)
12 tests passed (12 total in 1 test suite, run time 0.384s)
```

This isn't the nicest output though, if you run `npm run test -- --verbose` you should see a much nicer list of tests that passed like so:

```
PASS  src/__tests__/actions.test.js (0.364s)
 actions
   changeLocation
     ✓ it should have a type of CHANGE_LOCATION (5ms)
     ✓ it should pass on the location we pass in (1ms)
   setSelectedDate
     ✓ it should have a type of SET_SELECTED_DATE (1ms)
     ✓ it should pass on the date we pass in
   setSelectedTemp
     ✓ it should have a type of SET_SELECTED_TEMP (1ms)
     ✓ it should pass on the temp we pass in
   setData
     ✓ it should have a type of SET_DATA
     ✓ it should pass on the data we pass in (1ms)
   setDates
     ✓ it should have a type of SET_DATES
     ✓ it should pass on the dates we pass in
   setTemps
     ✓ it should have a type of SET_TEMPS (1ms)
     ✓ it should pass on the temps we pass in

12 tests passed (12 total in 1 test suite, run time 0.392s)
```

And this is what your `actions.test.js` file could look like:

```
// actions.test.js
```

```
import {
    changeLocation,
    setSelectedDate,
    setSelectedTemp,
    setData,
    setDates,
    setTemps
} from '../actions';

describe('actions', function() {
    describe('changeLocation', function() {
        it('should have a type of CHANGE_LOCATION', function() {
            expect(changeLocation().type).toEqual('CHANGE_LOCATION');
        });

        it('should pass on the location we pass in', function() {
            var location = 'Vienna, Austria';
            expect(changeLocation(location).location).toEqual(location);
        });
    });

    describe('setSelectedDate', function() {
        it('should have a type of SET_SELECTED_DATE', function() {
            expect(setSelectedDate().type).toEqual('SET_SELECTED_DATE');
        });

        it('should pass on the date we pass in', function() {
            var date = '2016-01-01';
            expect(setSelectedDate(date).date).toEqual(date);
        });
    });

    describe('setSelectedTemp', function() {
        it('should have a type of SET_SELECTED_TEMP', function() {
            expect(setSelectedTemp().type).toEqual('SET_SELECTED_TEMP');
        });

        it('should pass on the temp we pass in', function() {
            var temp = '31';
            expect(setSelectedTemp(temp).temp).toEqual(temp);
        });
    });

    describe('setData', function() {
        it('should have a type of SET_DATA', function() {
            expect(setData().type).toEqual('SET_DATA');
        });


        it('should pass on the data we pass in', function() {
            var data = { some: 'data' };
            expect(setData(data).data).toEqual(data);
        });
    });

    describe('setDates', function() {
        it('should have a type of SET_DATES', function() {
            expect(setDates().type).toEqual('SET_DATES');
        });

        it('should pass on the dates we pass in', function() {
            var dates = ['2016-01-01', '2016-01-02'];
            expect(setDates(dates).dates).toEqual(dates);
        });
    });

    describe('setTemps', function() {
        it('should have a type of SET_TEMPS', function() {
            expect(setTemps().type).toEqual('SET_TEMPS');
        });

        it('should pass on the temps we pass in', function() {
            var temps = ['31', '32'];
            expect(setTemps(temps).temps).toEqual(temps);
        });
    });
});
```

Perfect, that part of our app is now comprehensively tested and we’ll know as soon as somebody breaks something! Onwards to the reducer!