Test Driven Development Part 2

A first look at TDD for React

Manulife University Engineering

February 10, 2020

Manulife

TDD - Part 2 Agenda

React Application

- Download from git
- Setup & npm install
- Test

State management

- Clicking
- onChange
- Routing

Testing Redux

- Action Creator
- Reducer
- Store
- Container

Component management

- Shallow, mount
- Snapshots

Notes:

- Components
- Snapshots
- Shallow, mount
- Reports
- Enzyme vs React Testing Library

React Applications

Manulife

Download & setup

Git → mu-materials

Download from git:

https://git.platform.manulife.io/mu-materials/tdd-uu-exercises

OR:

ssh://git@git.platform.manulife.io:2222/mu-materials/tdd-uu-exercises.git

There are 2 applications:

\tdd-uu-exercises\tdd-doctors-patients-react-redux

\tdd-uu-exercises\tdd-uu-jest-enzyme

You will need to 'cd' into both folders and perform 'npm install'.

Manulife

State Management

State Management - Hands-on Exercise

- Use the following exercise:
  - Tdd-uu-jest-enzyme

State change - Clicking

Do not test the state → state(); instead test state change!

- Write a test to test a counter that increments a value when a button is clicked
- The test should 'simulate' clicking the button to increment the value and then test that the value is incremented by 1
- Implement: Create a Counter class that satisfies the test described

- Note: use your tests to verify the **behaviour** and not the *implementation*

Manulife

State change – onChange

Testing changes on the page based on user input/action

- Write tests to test input fields
- Make a change to the default input and check that it’s changed

- Test both the default input and the updated input

Manulife

Route Management

Route Management - Hands-on Exercise

- Use the following exercise:
  - Tdd-uu-jest-enzyme

Routing

Test 2 instances of routes

- Write tests to test that when the ‘root’ route (‘/’) is called, it redirects the user to the proper HomePage
- Write tests to test that when the ‘doctors’ route (‘/doctors’) is called, it redirects the user to the proper DoctorsPage
- Write tests to test that when the ‘patient’ route (‘/patient’) is called, it redirects the user to the proper PatientPage
- Write tests to test that when any other random route is called, it redirects to the user to the NotFoundPage!
- Note: you need to use MemoryRouter to encapsulate your main page before routing

Manulife

Testing Redux

Testing Redux - Hands-on Exercise

- Use the following exercise:
  - Tdd-doctors-patients-react-redux

Action Creators

Action creators are pure functions

- Very simple to test
  - Given input, test output
- If dispatch was called explicitly, dispatch can be mocked and what's passed into dispatch call can be checked
- Call jest.clearAllMocks() within afterEach() call as a good practice
- redux-mock-store & fetchMock can be used to test async calls with thunk

Manulife

Reducers

Reducers are pure functions too

- Very simple to test
  - Given sample input, test output
- Use toEqual() rather than toBe() since it is recommend to follow immutability pattern

Manulife

Store

Great place to do Integration tests

- Mostly boilerplate code

Manulife

Container

Testing redux connected components

- Name export component without changing default exported component
- mapStateToProps & mapDispatchToProps are pure functions as well

    - multiple ways of testing them

    - simplest way is to export them, but we expose the private members to the outside world

    - another option is to mock/spy on the function, and verify it get called by simulating user's interaction

- Lifecycle events can be triggered manually if needed

Manulife

Testing Components

Testing Components - Hands-on Exercise

- Use the following exercise:
  - Tdd-doctors-patients-react-redux

Shallow, mount, render

Requirement: Create a list to render items pass through in an array

- Write test cases before writing code
- Use proper ”describe”, “it” and ”expect”
- Remember to use props for passing data
- Use <shallow/mount> to test
- Use wrapper to find elements
- Use wrapper.debug() to see different rendered objects

Manulife

Snapshot

Requirement: take a backup of a component's state

- Write a test to take a snapshot of a component using renderer
- View the "snapshot"
- Change state of component
- Re-run test case to ensure it fails because the state is now different
- Change state back to original
- Re-run test case to pass

- Note: use the wrapper debug method to show what's "wrapped"

Manulife

Final Notes

UI testing rules

One component should have only one snapshot.

Testing props:

- test the default
- change value and re-test

Event testing:

- mock event → simulate it → expect event was called
- mock event → simulate event with params → expect event called with params
- pass necessary props → render component → simulate event → expect certain behavioiur on called event

Testing conditions:

- test if-else conditions
- test empty/edge cases

States testing:

- check current state
- Check state after calling event

Manulife

UI testing rules con’t

Shallow vs mount

Shallow: Constructor → render → componentDidMount

Mount: Full render include child components → requires DOM → execution time more costly

Render: Calls renders → renders all children

Rule of thumbs:

Always begin with **shallow**.

If you want to test children behavior, use **mount**.

If you want to test children rendering with less overhead than mount and you are not interested in lifecycle methods, use **render.**

UI testing rules con’t

When NOT to use snapshots

Third-party libraries:

- don’t test functionality taken from another library

Constants:

- these are not changeable

Things not related to the tested component!

When something is different every time you test it

- utility function that returns a random string every time
- New components, wait until it’s polished

Manulife

Reports

Coverage report for your UI unit testing

> Npm run test:report

Manulife

Enzyme vs. React Testing Library

Which one to use?

Both are popular and powerful tools with great documentations

Very different mindset

- Enzyme enable us to access the interval workings of components
- RTL is very user behavior focused

Enzyme has a lot of open issues since day 1 (buggy ☹)

Manulife

Manulife