**LECTURE 2 – Versions used in this Course**

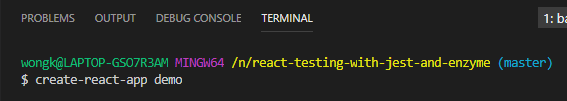
* These are the following versions used in this tutorial:
* [**create-react-app@1.5.2**](mailto:create-react-app@1.5.2)
* [**node@8.2.1**](mailto:node@8.2.1)
* [**npm@5.3.0**](mailto:npm@5.3.0)
* [**react@16.3.2**](mailto:react@16.3.2)
* [**jest@22.4.3**](mailto:jest@22.4.3)
* [**enzyme@3.3.0**](mailto:enzyme@3.3.0)
* [**Enzyme-adapter-react-16@1.1.1**](mailto:Enzyme-adapter-react-16@1.1.1)
* [**redux@4.0.0**](mailto:redux@4.0.0)
* [**React-redux@5.0.7**](mailto:React-redux@5.0.7)
* [**Redux-thunk@2.2.0**](mailto:Redux-thunk@2.2.0)
* [**axios@0.18.0**](mailto:axios@0.18.0)
* [**moxios@0.4.0**](mailto:moxios@0.4.0)

**LECTURE 3 – TDD: What and Why**

* In this lecture we will be implementing **TDD** which means that we will be writing the tests before we write the code.
* You write a shell version of the code, enough to call it from the test, then you write a test that fails (We do this to make sure the test fails before they succeed), then you write the code that makes the tests past.
* TDD helps you write better code.
* The code is better organized.
* The code is more test-able.
* There are fewer bugs.
* We will have **Regression Testing** built-in, meaning that all the tests that we’ve ever written will get run every time you make a change.

**LECTURE 4 – create-react-app**

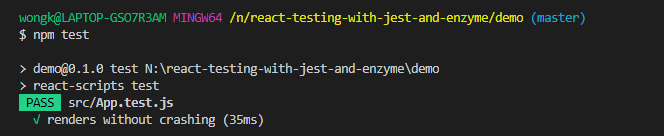
* We will now install the **jest** and **babel-jest** libraries by using **create-react-app**. The create-react-app library creates these 2 libraries automatically.
* If you don’t already have the **create-react-app** library installed then do that first.
* Open VS Code and open a terminal.
* In the terminal create your React App by typing **create-react-app demo**.



**LECTURE 5 – Setting up Jest without create-react-app**

**LECTURE 6 – Demo of Jest Watch Mode**

* Open your project at the root (demo folder) in VS Code.
* We are going to test Jest.
* In a terminal type **npm test**.
* A Jest Test will be executed.



* These Jest tests will be run each time you save the file.
* Go to **App.js** and make the following changes.

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

class App extends Component {

render() {

return (

<div className="App">

<h1>Hello World</h1>

<header className="App-header">

<img src={logo} className="App-logo" alt="logo" />

<p>

Edit <code>src/App.js</code> and save to reload.

</p>

<a

className="App-link"

href="https://reactjs.org"

target="\_blank"

rel="noopener noreferrer"

>

Learn React

</a>

</header>

</div>

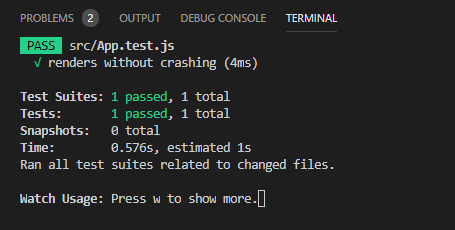
);

}

}

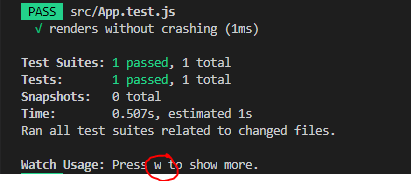
export default App;

* When you save the changes a Jest Test will automatically be run.

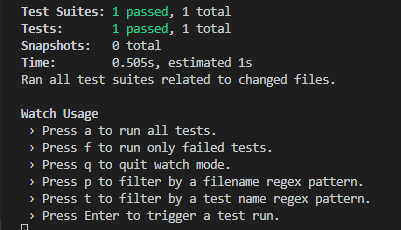


**LECTURE 7 – More about Jest Watch Mode, Test Files and Tests**

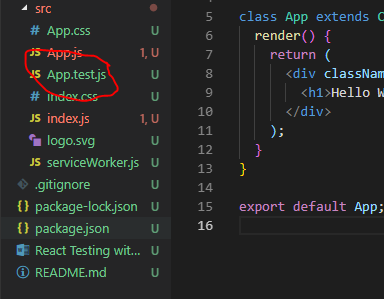
* Jest Watch Mode only watches for changes since your last commit.
* if you have already run **npm test** and then go to the terminal you should see the following screen.



* It says on the screen that if you press **2** you will get to see more. This will be the **Jest Watch Mode**. Press **w.**

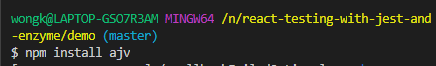


* Press **a** and it will run all tests.
* Press **f** to run only failed tests.
* All the other commands are listed above.
* Jest knows which test files to run because it runs all files that end in **.test.js**, such as **App.test.js**.



**LECTURE 8 – Enzyme Introduction and Setup**

* Enzyme is a tool that creates a virtual DOM for testing.
* Allows testing without a browser.
* Enzyme uses the create-react-dom under the hood but the enzyme interface gives us a much more extensive tool kit to work with our virtual dom.
* You can use jQuery selectors.
* Enzyme gives us access to **props** and **state**.
* We will now get started setting up Enzyme.
* First, install the **ajv** library by typing in the terminal **npm install ajv**.



* Next we will install **Enzyme** and an **Enzyme adaptor**.
* Type the following in the terminal: **npm install –save-dev jest enzyme enzyme-adapter-react-16** (we use enzyme-adapter-react-16) because we are using version 16 of react, which you can find in package.json).

**IMPORTANT:** You need to install [**jest@23.6.0**](mailto:jest@23.6.0) **otherwise it npm test doesn’t work**.



* We are now going to make the following changes to **App.test.js**.

import React from 'react';

import Enzyme from 'enzyme';

import EnzymeAdapter from 'enzyme-adapter-react-16';

import ReactDOM from 'react-dom';

import App from './App';

Enzyme.configure({ adapter: new EnzymeAdapter() });

it('renders without crashing', () => {

const div = document.createElement('div');

ReactDOM.render(<App />, div);

ReactDOM.unmountComponentAtNode(div);

});

**LECTURE 9 – Using Enzyme in a test**

* We will need to destruct the **shallow** function from the **enzyme** library.
* The shallow function wraps that code that you want to perform the tests on into an object.

import React from 'react';

import Enzyme, { shallow } from 'enzyme';

import EnzymeAdapter from 'enzyme-adapter-react-16';

import App from './App';

Enzyme.configure({ adapter: new EnzymeAdapter() });

it('renders without crashing', () => {

});

* Add the following shallow function and assign it to a constant called **wrapper**.

import React from 'react';

import Enzyme, { shallow } from 'enzyme';

import EnzymeAdapter from 'enzyme-adapter-react-16';

import App from './App';

Enzyme.configure({ adapter: new EnzymeAdapter() });

it('renders without crashing', () => {

const wrapper = shallow(<App />);

});

* We will now use the **expect** function.
* In **App.test.js**, write the following.

import React from 'react';

import Enzyme, { shallow } from 'enzyme';

import EnzymeAdapter from 'enzyme-adapter-react-16';

import App from './App';

Enzyme.configure({ adapter: new EnzymeAdapter() });

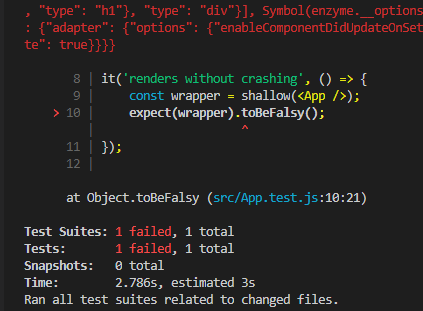
it('renders without crashing', () => {

const wrapper = shallow(<App />);

expect(wrapper).toBeTruthy();

});

* This test should pass.
* Now write a **toBeFalsy** method. This should fail.



**IMPORTANT: You can interchange “test” with “it” so the following 2 code are the same.**

**it('renders without crashing', () => {**

**const wrapper = shallow(<App />);**

**});**

**test('renders without crashing', () => {**

**const wrapper = shallow(<App />);**

**});**

**LECTURE 10 – Types of tests**

* **Unit Tests:** Tests one piece of code. These are very narrow and isolated in scope.
* **Integration Tests:** Tests how multiple units work together.
* **Acceptance / End-to-End (E2E) tests:** Tests how a user would actually interact with an app and see whether the user interactions work from end-to-end from User interface down to database and back again. Tend to use tools like **Selenium**.

**LECTURE 11 – Test Behaviour, not Implementation**

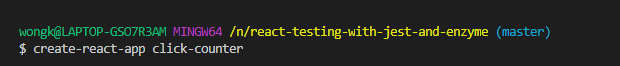
* A primary goal of testing is to test the behaviour, not the implementation.
* Want to make the tests robust enough, that any code refactors that don’t change the behaviour of the application should not affect the tests.
* Should not have to re-write the tests each time we refactor the code.

**LECTURE 12 – Snapshot Testing?**

* Snapshot Testing is a way to “freeze” a component.
* It freezes a set of code or output at a certain point-in-time, then the test will always compare any future output against that frozen output.
* The Test fails if there are any changes.
* **Snapshot Testing will not be used in this course because it can’t be used in Test Driven Development. These tests are quite brittle.**

**LECTURE 13 – Demo and Start Click Counter App**

* Now we will start by creating an app called **click-counter**.
* Open VS Code and in the terminal type **create-react-app click-counter**.



* We are now going to install **ajv** using **npm install ajv**.



* Now install **jest**, **enzyme** and **enzyme-adapter-react-16** by type the following in the terminal: **npm install –-save-dev** [**jest@23.6.0**](mailto:jest@23.6.0)[**enzyme@3.3.0**](mailto:enzyme@3.3.0) **enzyme-adapter-react-16@1.1.1**



**LECTURE 14 – Set up Enzyme and Write Tests**

* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf
* Asdf
* Asdf
* Asf
* Asdf
* Asf
* As
* Df
* Ads
* Fas
* Fsad
* Fa
* sdf