**React Testing Library**

**Jest vs RTL**

**Jest:**

* Jest is a JavaScript testing framework.
* Jest is a test runner that finds tests, runs the tests, determines whether the tests passed or failed and reports it back in a human readable manner.

**RTL**

* JavaScript testing utility that provides virtual DOM for testing React components.
* React testing library provides a virtual DOM which we can use to interact with and verify the behavior of a react component.
* Testing library is in fact a family of packages which helps test UI components.
* The core library is called DOM testing library and RTL is simply a wrapper around this core library to test React applications in an easier way.

**Types of Tests**

|  |  |
| --- | --- |
| Unit Tests | * Focus is on testing the individual building blocks of an application such as a class or a function or a component * Each unit or building block is tested in isolation, independent of other units * Dependencies are mocked * Run in a short amount of time and make it very easy to pinpoint failures * Relatively easier to maintain |
| Integrations Tests | * Focus is on testing a combination of units an ensuring they work together * Take longer than unit tests |
| E2E Tests | * Focus is on testing the entire application flow and ensuring it works as designed from start to finish * Involves in a real UI, a real backend database, real services etc. * Take the longest as they cover the most amount of code * Have a cost implication as you interact with real APIs that may charge based on the number of requests |

**What types of test are we writing?**

**Answer: RTL Philosophy**

* The more your tests resemble the way your software is used, the more confidence they can give you
* Tests we are going to learn to write in this series strike a balance b/w unit tests in the sense they are at a component level and easy to write and maintain and E2E tests in the sense thy resemble the way a user would interact with the component
* With React Testing Library (RTL) we are not concerned about the implementation of a component
* Instead we are testing how the component behaves when the user interacts with it
* RTL will not care if you add 4+4 or 5+3 to display the number 8
* Refactoring will not affect your tests as long as the end result is the same