Introduction to Jest

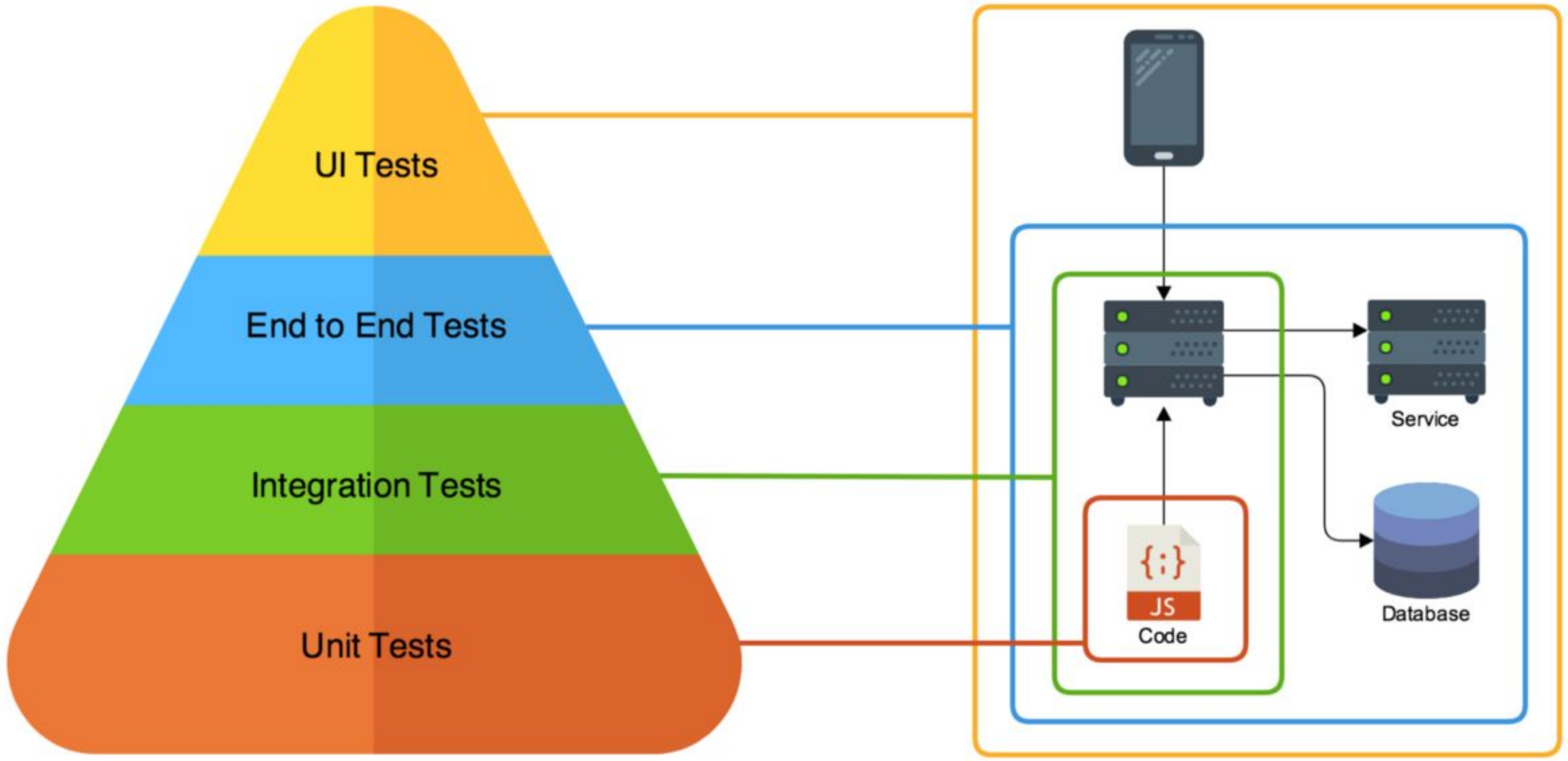
sebinbenjamin

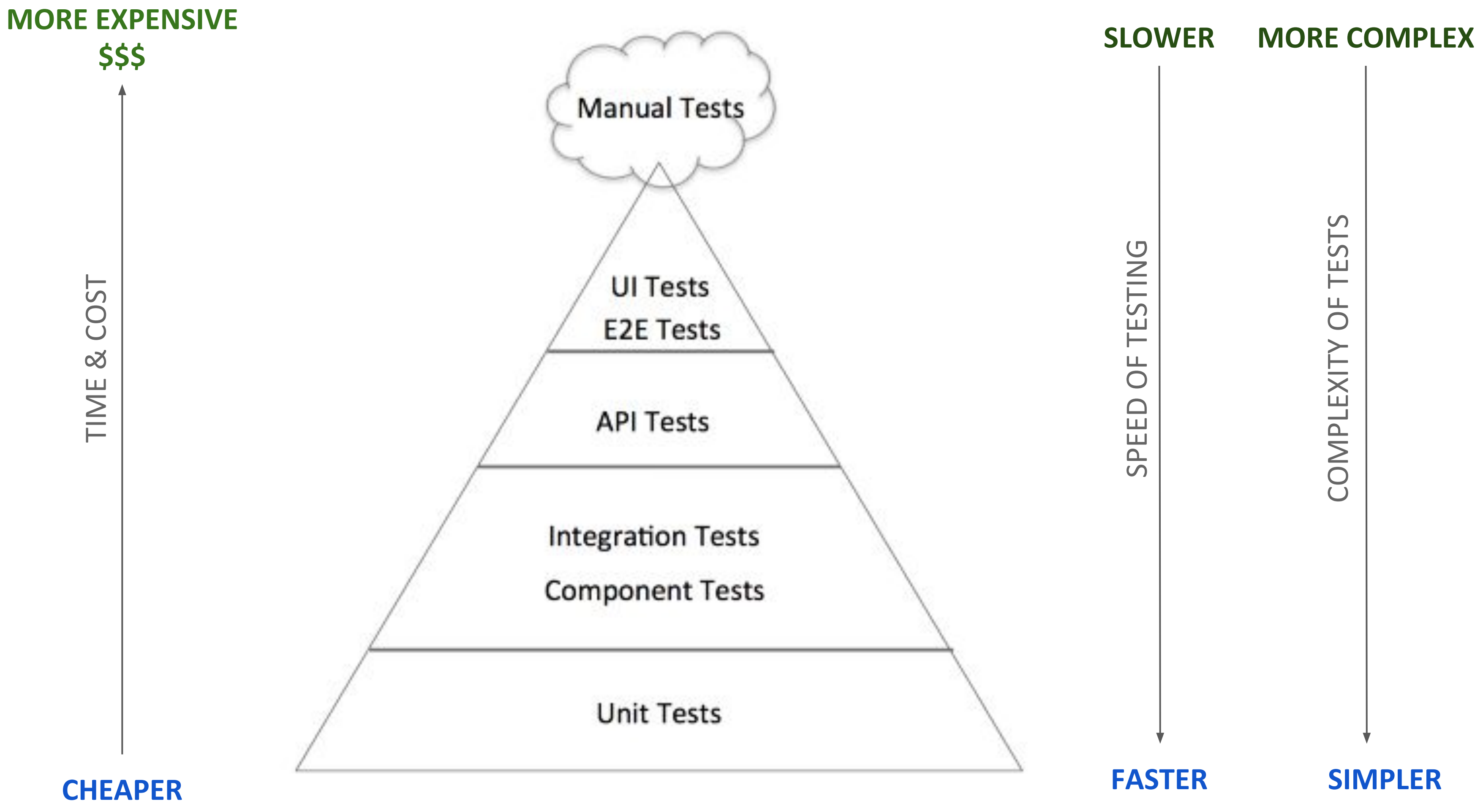
What we’ll look at

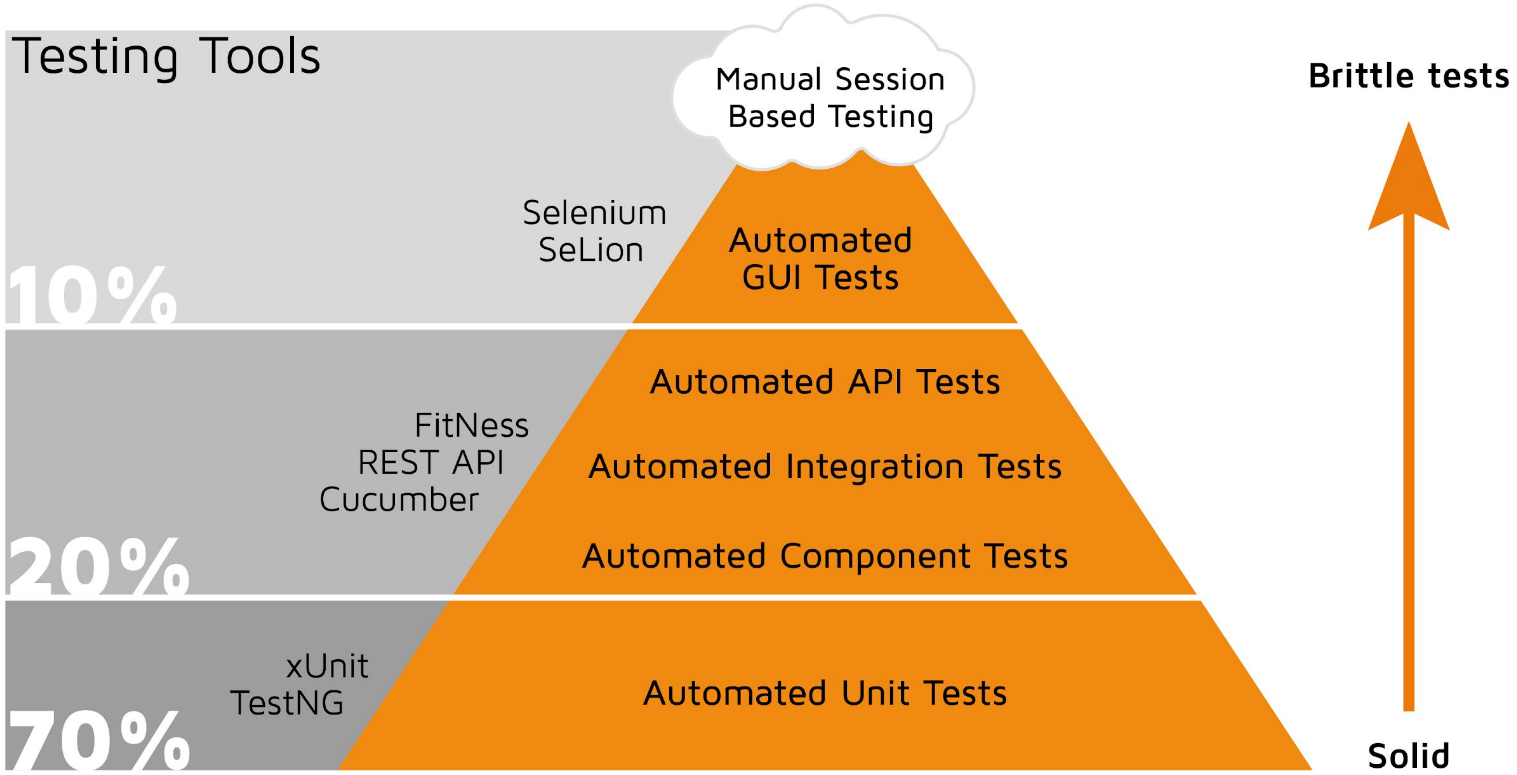
- Recap on Software Testing
  - The testing pyramid
  - What makes a good test
- What is Jest?
  - Jest Installation
- What is React Testing Library?
- Snapshot Testing

Recap on
Software Testing

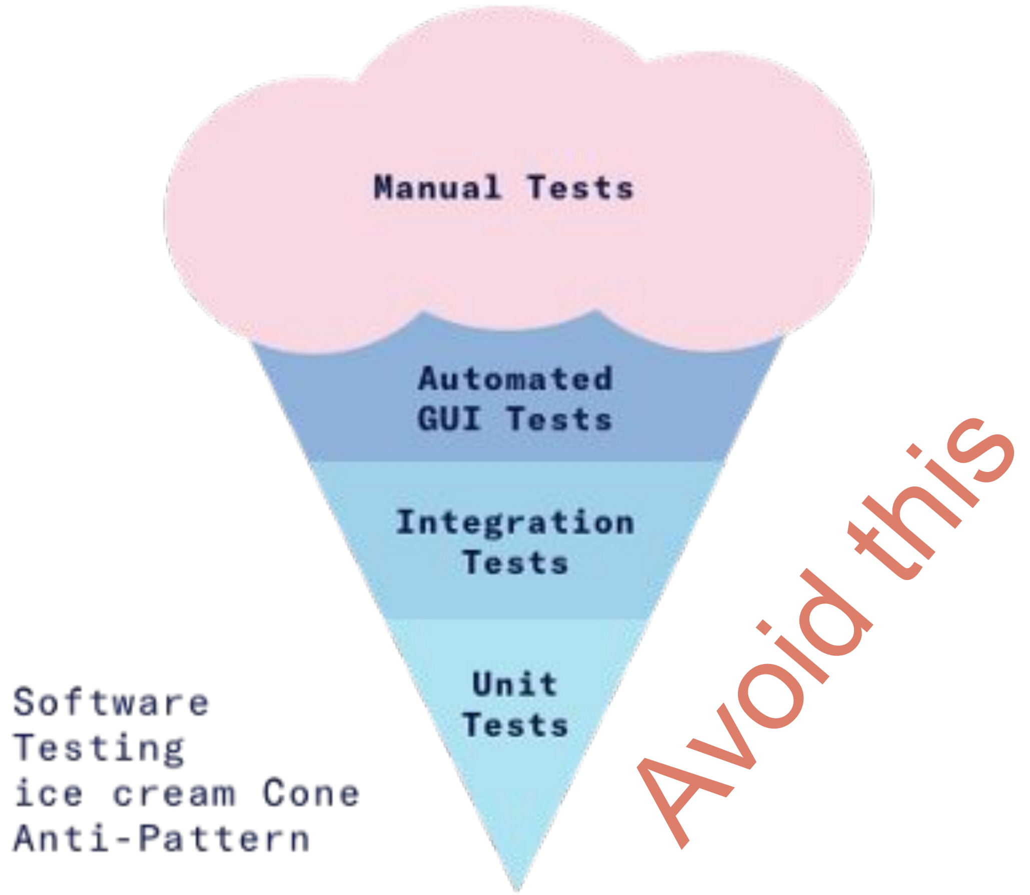
The Testing Pyramid




MORE EXPENSIVE
$$$
TIME & COST
CHEAPER
Manual Tests
UI Tests
E2E Tests
API Tests
Integration Tests
Component Tests
Unit Tests
SLOWER
SPEED OF TESTING
FASTER
MORE COMPLEX
COMPLEXITY OF TESTS
SIMPLER

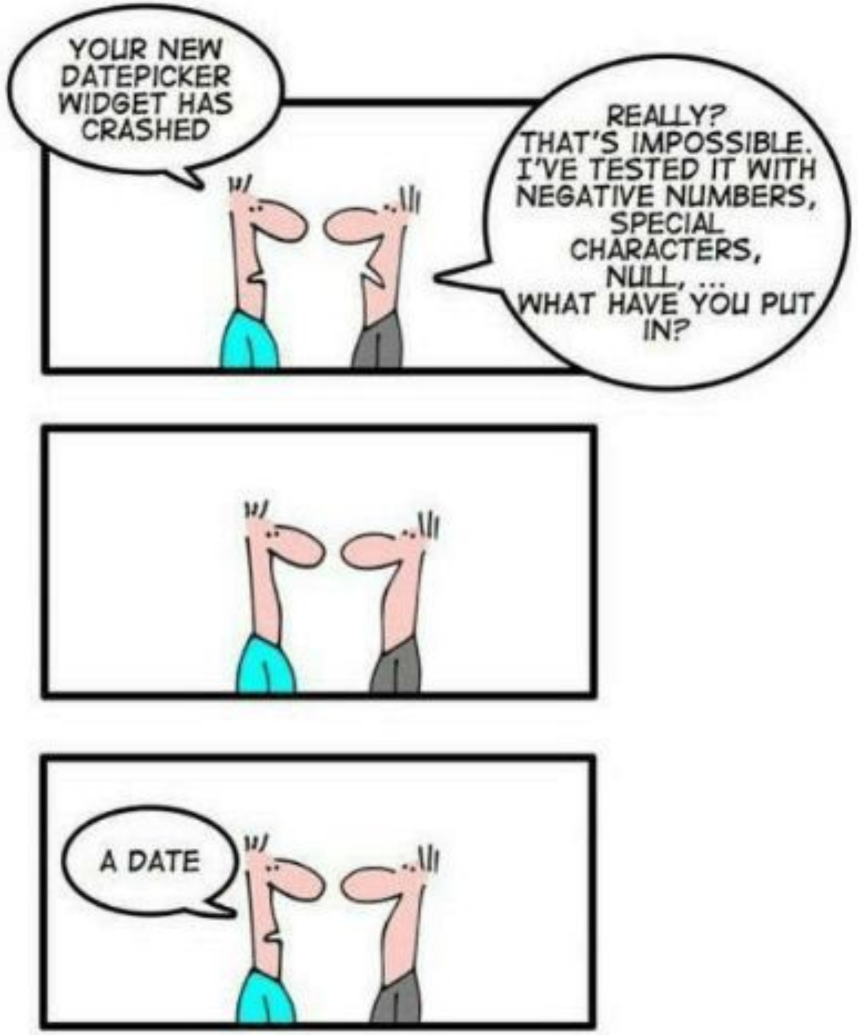

Testing Tools
Manual Session Based Testing
Brittle tests
Selenium
SeLion
Automated GUI Tests
10%
Automated API Tests
FitNess
REST API
Cucumber
Automated Integration Tests
Automated Component Tests
20%
xUnit
TestNG
Automated Unit Tests
70%
Solid

Icecream cone Anti-Pattern



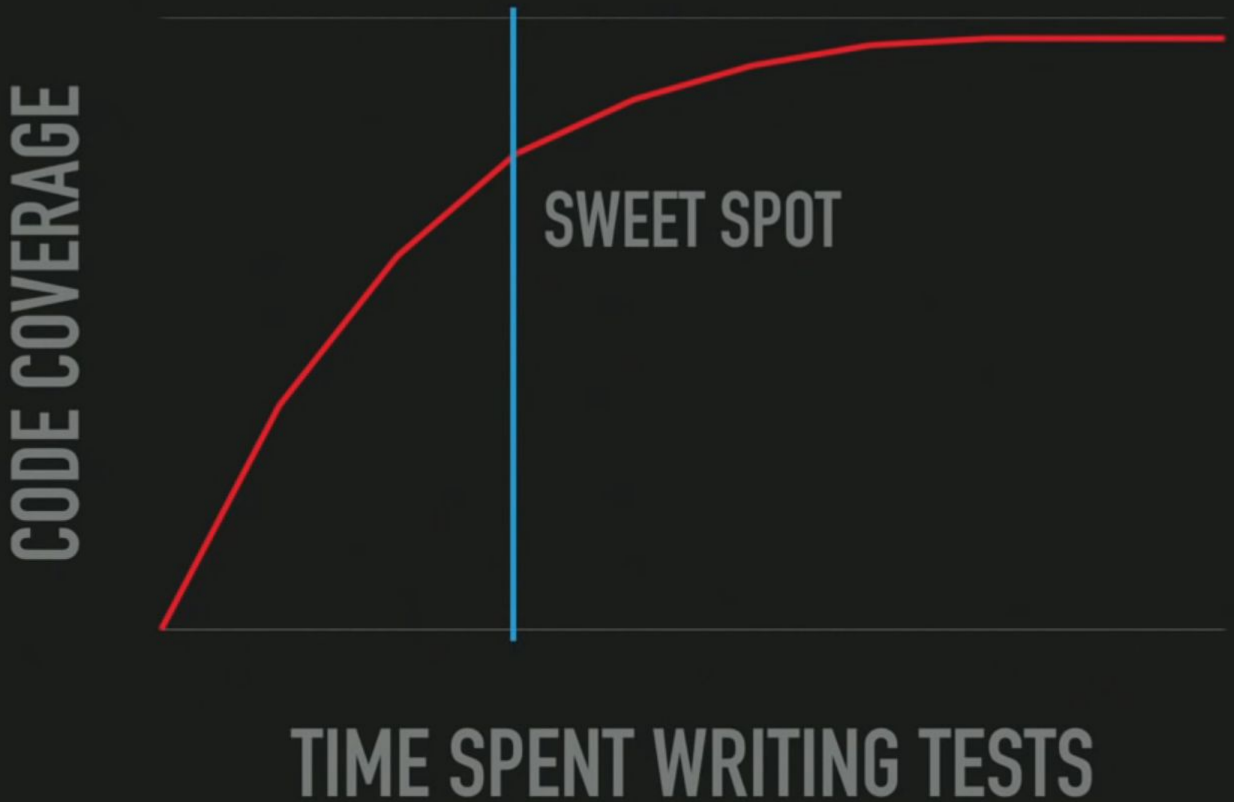
What makes a good test

- Runs fast
- Should not break often
- Should be easily read/understandable by others.
- Tests has to catch bugs.
- Good coverge to effort ratio

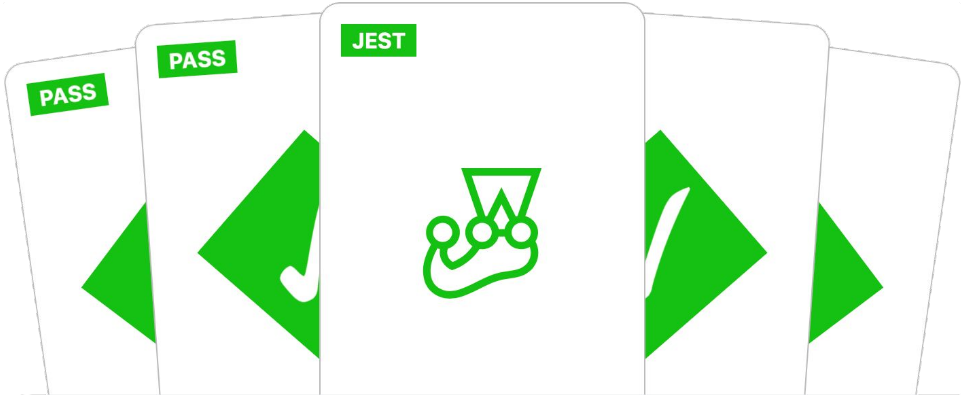

YOUR NEW DATEPICKER WIDGET HAS CRASHED
REALLY? THAT'S IMPOSSIBLE. I'VE TESTED IT WITH NEGATIVE NUMBERS, SPECIAL CHARACTERS, NULL, ... WHAT HAVE YOU PUT IN?
A DATE

https://me.me/i/your-new-datepi

TESTING ROI



JEST



Jest is a delightful JavaScript ***Testing Framework*** with a focus on simplicity.

Jest is a *JavaScript* ***test runner****,* that is, a JavaScript library for **creating, running, and structuring tests**.

Jest also contains ***assertion*** features, to verify that things are correct

Jest can also perform "***mocking***" by creating mock functions.

React Testing Library

Builds on top of DOM Testing Library by adding ***APIs for working with React components***.

Projects created with Create React App have out of the box support for React Testing Library.

React Testing Library is a very light-weight solution for testing React components. It provides light utility functions on top of react-dom and react-dom/test-utils.

What this library is NOT:

1. A test runner or framework
2. Specific to a testing framework

Install Jest

```
npm install --save-dev jest
```

Install React Testing Library

```
npm install --save-dev
@testing-library/react
```

Run Tests

```
npm run test
npm run test -- -t ‘test_description’
```

File matching for the Test runner

- Any files with a suffix of **`.test`** or **`.spec`** (e.g. Component.test.js or Component.spec.js).
- By default it looks for `.js`, `.jsx`, `.ts` and `.tsx` files inside of **`__tests__`** **folders**.

https://jestjs.io/docs/en/api

`describe(name, fn)`

`describe(name, fn)` creates a block that groups together several related tests. Basically breaks your test suite into components

`test(name, fn, timeout?)` same as **`it(name, fn, timeout?)`**

All you need in a test file is the test method which runs a test.

- The first argument is the test name
- The second argument is a function that contains the expectations to test.
- The third argument (optional) is timeout (in milliseconds) for ***specifying how long to wait before aborting***. Note: The default timeout is 5 seconds.

`beforeAll(fn, timeout?)`

Runs a function before any of the tests in this file run. This is often useful if you want to set up some global state that will be used by many tests.

Optionally, you can provide a timeout (in milliseconds) (default 5 seconds) for specifying how long to wait before aborting.

`afterAll(fn, timeout)`

Runs a function after all the tests in this file have completed. This is often useful if you want to clean up some global setup state that is shared across tests.

Optional timeout, similar to beforeAll()

Snapshot Testing

Tool to make sure your UI does not change unexpectedly.

A snapshot test case ***renders a UI component***, ***takes a snapshot***, then ***compares it to a reference snapshot*** file stored alongside the test.

The test will fail if the **two snapshots do not match**:

- Either the change is unexpected,
- The reference snapshot needs to be updated to the new version of the UI component.

```
npm run test -- -u
```

https://jestjs.io/docs/en/snapshot-testing

Thank you