

**Date :** 27 - 07 - 2020

**Morning Session :** 9 am – 11.00 PM

**By ~** Sundeep Charan Ramkumar Today

## **Topics:** Unit, Integration, E2E Testing

### Unit:

Verify that individual, isolated parts work as expected.

### Integration:

Verify that several units work together in harmony.

### End to End:

A helper robot that behaves like a user to click around the app and verify that it functions correctly. Sometimes called "functional testing" or e2e.

<https://medium.com/@rossbulat/testing-in-react-with-jest-and-enzyme-an-introduction-99ce047dfcf8>

<https://jestjs.io/docs/en/tutorial-react>

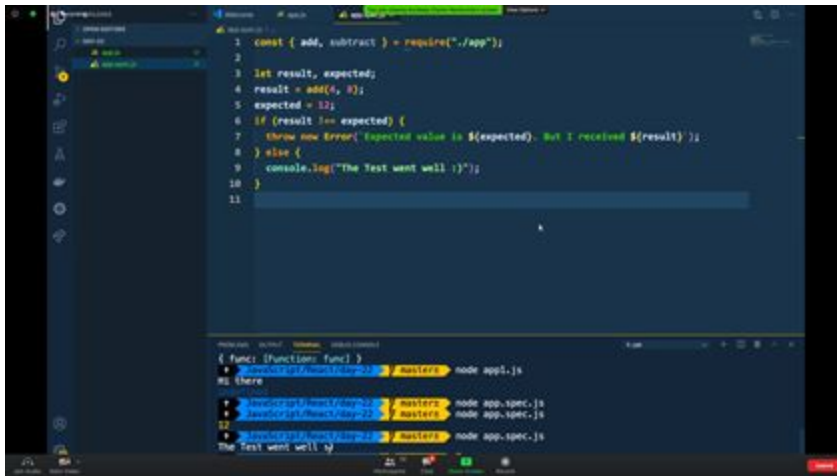
### Jargons:

- Test Runner → Mocha, Jasmine, Jest etc
- Assertion Library → Expected Value Checking → Chai, Enzyme, Jest
- Headless Browser → Puppeteer, Cypress, Sinon

We are going to focus on Jest

- Jest is a Terminal based
- Jest is a delightful JavaScript testing framework with a focus on simplicity.
- It can be installed with npm or Yarn. Jest fits into a broader category of utilities known as test runners.

## Simplest Test demo:

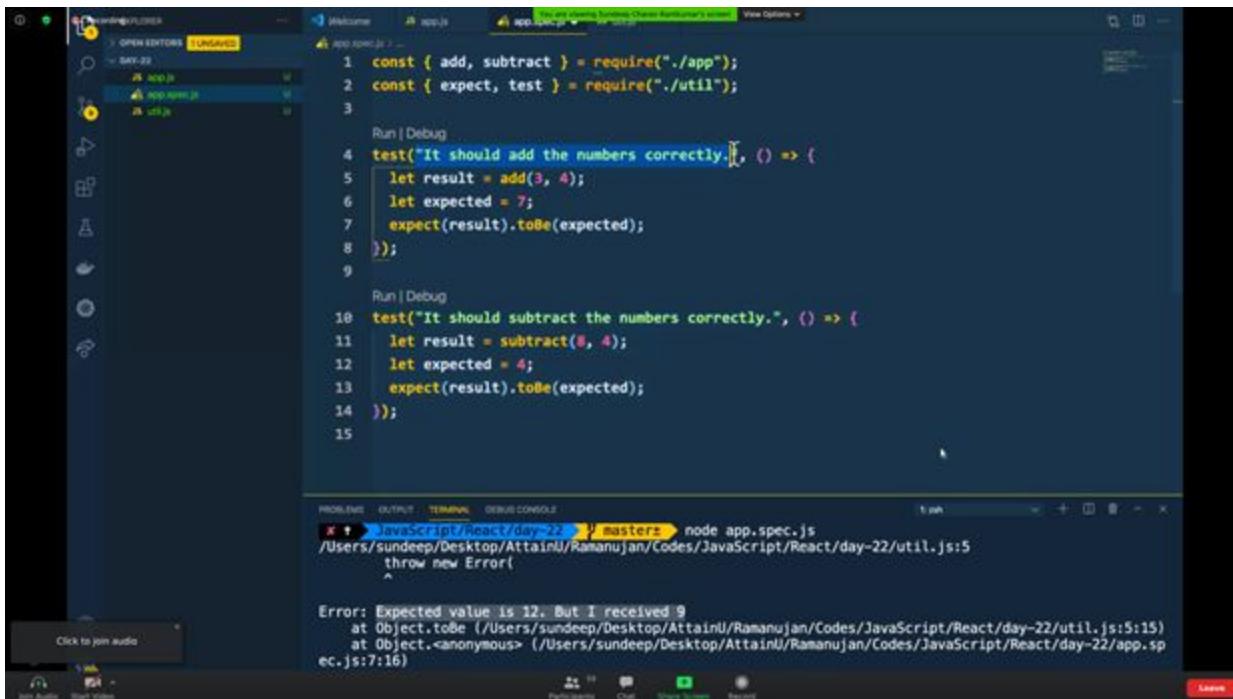


```
1 const { add, subtract } = require("./app");
2
3 let result, expected;
4 result = add(4, 4);
5 expected = 12;
6 if (result !== expected) {
7   throw new Error(`Expected value is ${expected}. But I received ${result}`);
8 } else {
9   console.log("The Test went well :)");
10 }
11
```

Terminal output:

```
{ func: [Function: func] }
node app.js
node app-spec.js
node app-spec.js
node app-spec.js
The Test went well :)
node app-spec.js
```

## Elegant Testing Demo example:



```
1 const { add, subtract } = require("./app");
2 const { expect, test } = require("./util");
3
4 test("It should add the numbers correctly.", () => {
5   let result = add(8, 4);
6   let expected = 7;
7   expect(result).toBe(expected);
8 });
9
10 test("It should subtract the numbers correctly.", () => {
11   let result = subtract(8, 4);
12   let expected = 4;
13   expect(result).toBe(expected);
14 });
15
```

Terminal output:

```
node app.spec.js
/Users/sundeep/Desktop/AttainU/Ramanujan/Codes/JavaScript/React/day-22/util.js:5
  throw new Error(
        ^
Error: Expected value is 12. But I received 9
    at Object.toBe (/Users/sundeep/Desktop/AttainU/Ramanujan/Codes/JavaScript/React/day-22/util.js:5:15)
    at Object.<anonymous> (/Users/sundeep/Desktop/AttainU/Ramanujan/Codes/JavaScript/React/day-22/app.spec.js:7:16)
```

## Jest Testing:

### Documentation :

<https://jestjs.io/>

## Installation:

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

In your `package.json` file, create a new npm script which runs Jest:

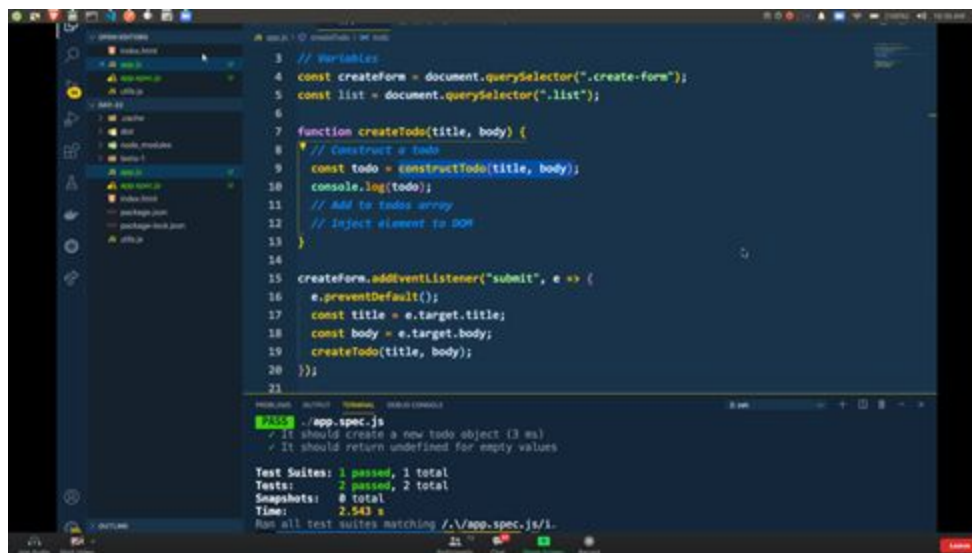
```
{
  ...
  "scripts": {
    "start": "webpack-dev-server --config ./webpack.config.js --mode de
    "test": "jest"
  },
  ...
}
```

In addition, we want to have more configuration in our tests written with Jest. Hence, pass an additional Jest configuration file to your Jest script:

```
{
  ...
  "scripts": {
    "start": "webpack-dev-server --config ./webpack.config.js --mode de
    "test": "jest --config ./jest.config.json"
  },
  ...
}
```

## Trying out Jest:

- Implemented jest to a Todo app ((Parcel was used here))



## Installing Puppeteer and using chromium automation:

Unit testing is a method where you check for a particular smallest functionality. True or False?

Attempted  
- 30  
(90.91%)

EASY



☒ True

90%

☐ False

10%

Chat (Alt+H)

What is the correct way to assert a reference type?

Attempted - 31 (93.94%)

EASY



☐ toBe

61.29%

☒ toEqual

38.71%

To launch a page in Puppeteer, which method is valid?

Attempted - 33 (100%)

EASY



☐ browser.go

24.24%

☒ browser.goTo

75.76%

What is the correct way to assert a reference type?

Attempted - 36 (105.88%)

EASY



☐ toBe

72.22%

☒ toEqual

27.78%

## Resources:

<https://www.youtube.com/watch?v=D9DdY2WmM-s>