

VUE.JS

Testing

- 1) Unit Testing ✓ → confidence in our code
- 2) Component Testing
- 3) End-to-End Testing

* Choosing our Unit Testing Framework

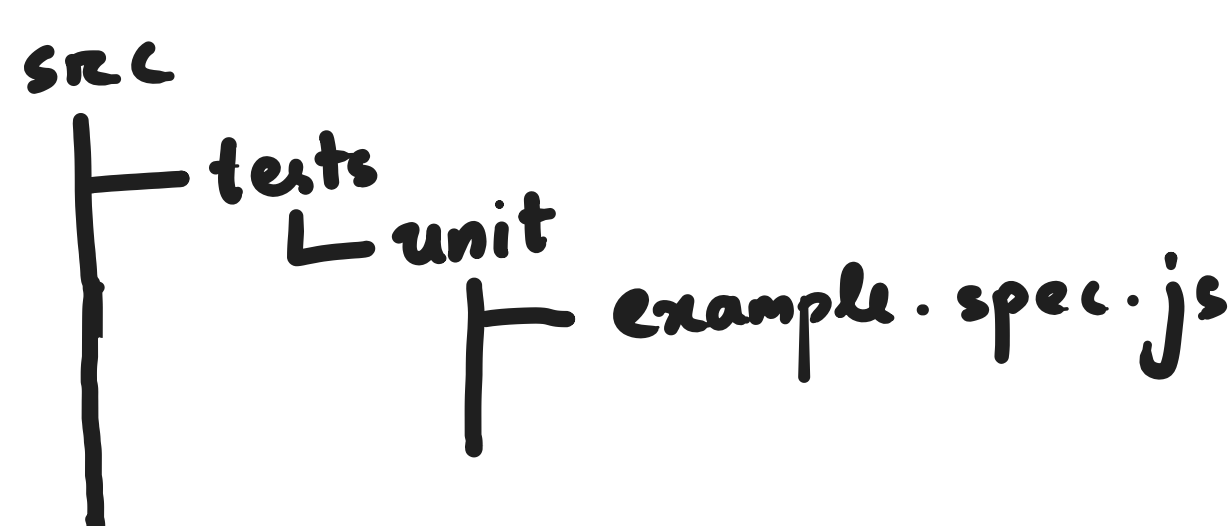
- 1) Jest (simplicity, Ability to snapshot of tests)
- 2) Mocha + Chai (Flexibility, Allow to choose different libraries, can execute tests in the browser)

↳ Things to consider:-

- i) First-class error reporting
- ii) Active community & team

1) Jest

Test cases are inside:-



Jest will pick up *.test.js or *.spec.js files in our project folder.

There are three fundamental methods in Jest:

- describe()
- test()
- expect()

i) describe():-

It is used to group related test cases together.

Takes two arguments: a string describing the group of tests
a callback contains the test cases

Example:-

```
describe('math', () => {
  test('2 + 2 should equal 4', () => {
    expect(2 + 2).toBe(4);
  })
})
```

ii) test():- ('it() is an alias for the 'test()' method)

```
test('2 + 2 should equal 4', () => {
  expect(2 + 2).toBe(4);
})
```

'test()' is used to define an individual test case.

It takes two arguments:-

- the first string argument, which describes the test case
- the second argument is a callback function which contains the code to be tested.

iii) expect:-

```
expect(2 + 2).toBe(4);
```

It is used to define the expected outcome of the test case

The argument is the value 2+2, which is expected to be equal to 4.

Example of test() with multiple expect() statements:-

```
test('compare function returns 1 if the first number is greater, 0 if both the numbers are the same, and -1 if the first number is smaller', () => {
  expect(compare(3, 2)).toBe(1);
  expect(compare(2, 3)).toBe(-1);
  expect(compare(3, 3)).toBe(0);
});
```

* Unit Testing an Axios Request

'axios-mock-adapter'
+
Jest

```
describe('Axios GET request', () => {
  it('returns the expected data', async () => {
    const mock = new MockAdapter(axios);
    const expectedData = { name: 'Akash Das', age: 30 };
    mock.onGet('/api/users').reply(200, expectedData);

    const response = await axios.get('/api/users');
    expect(response.data).toEqual(expectedData);
  });
});
```

* Component Testing

✓ 1) Vue Testing Library (@testing-library/vue)

2) Vue Test Utils

* End-to-End (E2E) Testing

→ Cross-browser testing
↳ Selenium
↳ LogRocket

→ Feedback loops (CI/CD pipelines)

→ Visibility in headless mode

1) Cypress.io

2) Nightwatch.js

3) Puppeteer

4) TestCafe