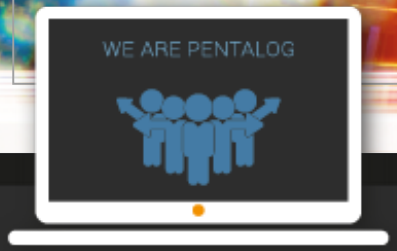
Pentalog

A FULL-STACK NEARSHORE IT GROUP SERVING COMPANIES
FROM STARTUPS TO FORTUNE 500s

Testing Vue Applications

Koida Leonid



WHO AM I

Koida Leonid	Web Developer, tango dancer, drummer
Linkedin	www.linkedin.com/in/leonidkoida
Main Technologies	JavaScript, PHP

What the testing is?

DEFINITION

A simple definition of testing is that testing an application is *the process of checking that an application behaves correctly*.

BRIEF HISTORY OF TESTING

Until 1956	Debugging oriented period
1957-1978	Demonstration oriented period
1979-1982	Destruction oriented period
1983-1987	Evaluation oriented period
From 1988	Prevention oriented period

Classification by D.Gelperin and W.C. Hetzel

TESTING METODOLOGIES(SOME OF THEM)

Component testing aka Unit tesing

Integration testing

System testing

Acceptance testing

COMPONENT TESTING (UNIT TESING)

Advantages

1- Unit testing finds problems early in the development cycle.

2 - In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written.

3 -Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly

4 - Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach.

COMPONENT TESTING (UNIT TESING)

Limitations and disadvantages

1 - Testing will not catch every error in the program

2 - Software testing is a combinatorial problem

3 - Another challenge related to writing the unit tests is the difficulty of setting up realistic and useful tests

JAVASCRIPT TESTING FRAMEWORKS

Jasmine

Unit.js

Karma

Jest

JSUnit

AVA

Mocha

QUnit

Why Jest?

1 - zero config

Jest aims to work out of the box, config free, on most JavaScript projects.

2 - snapshots

Make tests which keep track of large objects with ease. Snapshots live either alongside your tests, or embedded inline.

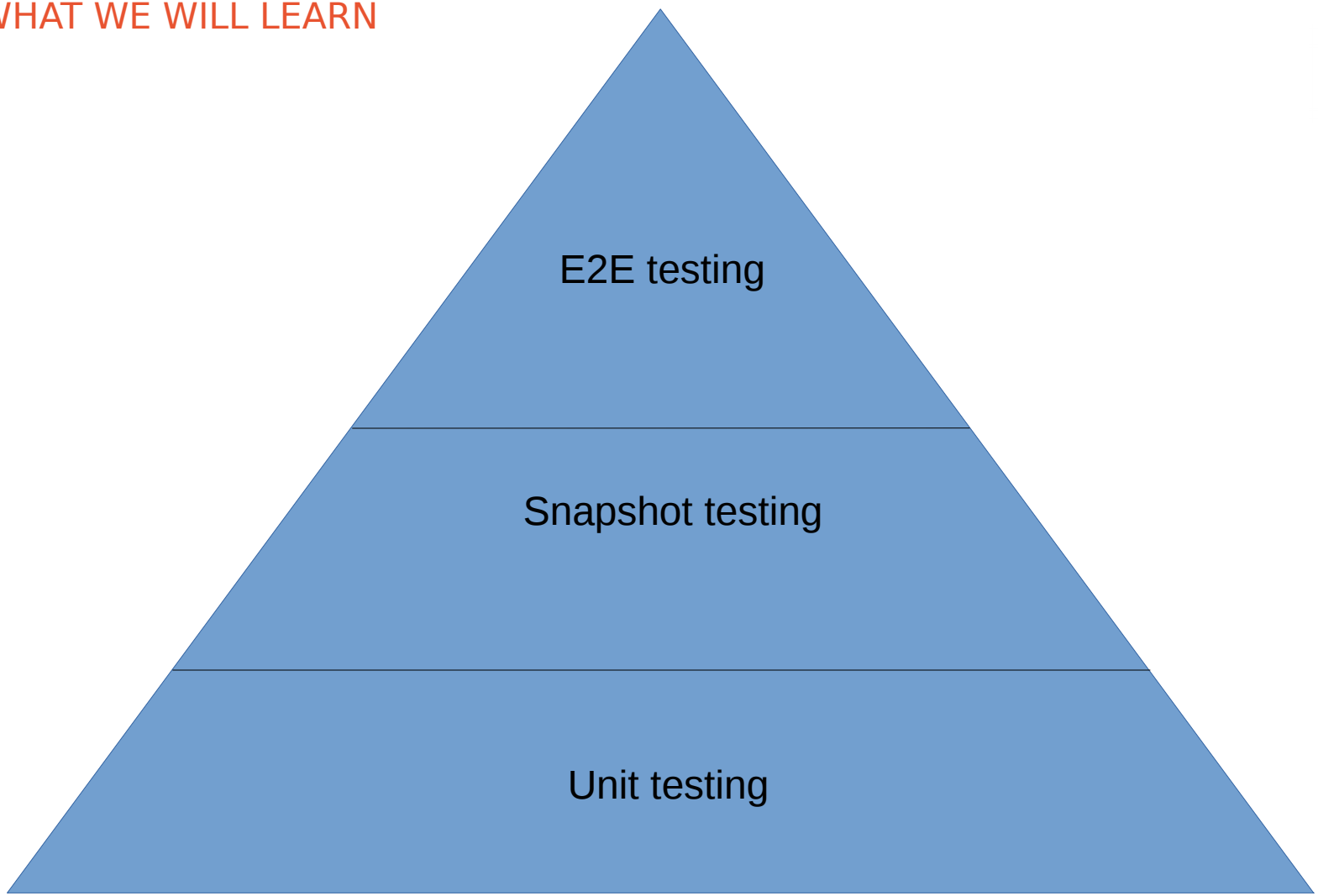
3 - isolated

Tests are parallelized by running them in their own processes to maximize performance.

4 - great api

From it to expect - Jest has the entire toolkit in one place. Well documented, well maintained, well good.

WHAT WE WILL LEARN



JEST AS A TESTING FRAMEWORK

Example of simple unit test

```
1  const shoppingList = [
2    'diapers',
3    'kleenex',
4    'trash bags',
5    'paper towels',
6    'beer',
7  ];
8
9  test('the shopping list has beer on it', () => {
10   expect(shoppingList).toContain('beer');
11   expect(new Set(shoppingList)).toContain('beer');
12 });
13
14 function compileAndroidCode() {
15   throw new ConfigError('you are using the wrong JDK');
16 }
17
18 test('compiling android goes as expected', () => {
19   expect(compileAndroidCode).toThrow();
20   expect(compileAndroidCode).toThrow(ConfigError);
21
22   // You can also use the exact error message or a regexp
23   expect(compileAndroidCode).toThrow('you are using the wrong JDK');
24   expect(compileAndroidCode).toThrow(/JDK/);
25 });
26
```

JEST AS A TESTING FRAMEWORK

Example of the snapshot test

```
import React from 'react';
import Link from '../Link.react';
import renderer from 'react-test-renderer';

it('renders correctly', () => {
  const tree = renderer
    .create(<Link page="http://www.facebook.com">Facebook</Link>)
    .toJSON();
  expect(tree).toMatchSnapshot();
})
```

```
exports[`renders correctly 1`] = `
<a
  className="normal"
  href="http://www.facebook.com"
  onMouseEnter={[Function]}
  onMouseLeave={[Function]}
>
  Facebook
</a>
`;
```

JEST AND VUE

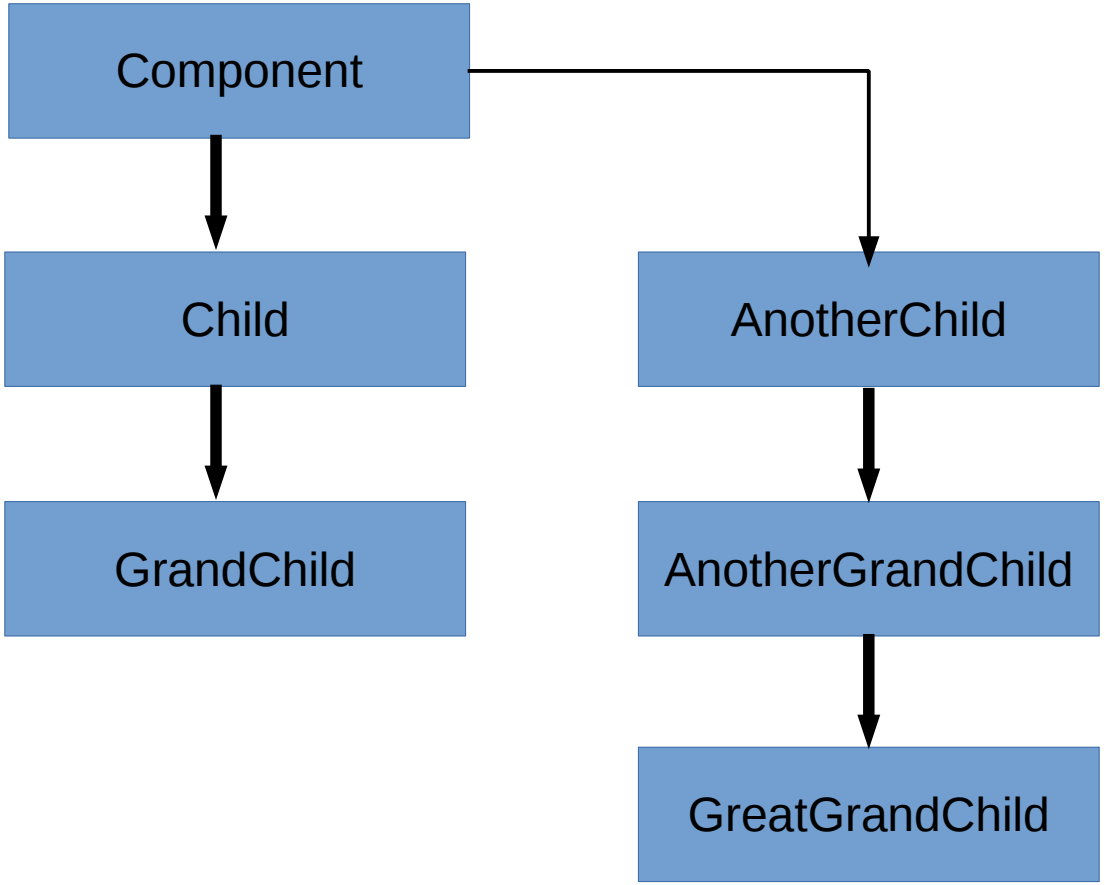
```
import Item from '../Item.vue'
import Vue from 'vue'

describe('Item.vue', () => {
  test('renders "item"', () => {
    const Ctor = Vue.extend(Item)
    const vm = new Ctor().$mount()
    expect(vm.$el.textContent).toContain('item')
  })
})
```

JEST, VUE AND VUE-TEST-UTILS

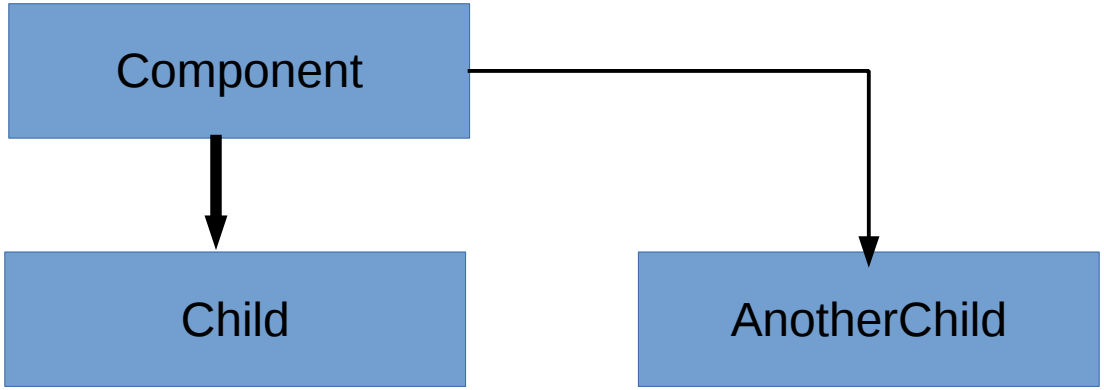
```
import { mount } from '@vue/test-utils'
import Item from '../Item.vue'

describe('Item.vue', () => {
  test('renders item', () => {
    const wrapper = mount(Item)
    expect(wrapper.vm.$el.textContent).toContain('item')
  })
})
```

Mount vs shallowMount

Mount


Component
Child
GrandChild
AnotherChild
AnotherGrandChild
GreatGrandChild

shallowMount



Let’s code

Links

Theory

https://en.wikiversity.org/wiki/Software_testing/History_of_testing
https://en.wikipedia.org/wiki/Component_testing
https://en.wikipedia.org/wiki/Integration_testing
https://en.wikipedia.org/wiki/System_testing
https://en.wikipedia.org/wiki/Acceptance_testing
https://en.wikipedia.org/wiki/Extreme_programming
https://en.wikipedia.org/wiki/Test-driven_development

Practice

https://jestjs.io/docs/en/getting-started
https://jestjs.io/docs/en/api
https://vue-test-utils.vuejs.org/guides/getting-started.html
https://vue-test-utils.vuejs.org/api/
https://vue-test-utils.vuejs.org/api/wrapper/#properties
https://vue-test-utils.vuejs.org/api/selectors.html
https://www.npmjs.com/package/@vue/cli-plugin-unit-jest