

단위 테스트



❖ 애플리케이션이 복잡해지면 자동화된 테스트 필요

- 기능의 추가, 변경시 새로운 오류가 발생하지 않는지 테스트
- 코드의 정상 작동을 빠르게 확인
- 예상되는 동작, 기능을 동적으로 문서화하고 발생하는 문제를 신속히 발견

❖ Vue-CLI가 지원하는 템플릿에 테스트 기능 포함

- Karma + Mocha + Chai
 - Mocha : 자바스크립트 테스트 프레임워크
 - Karma : 프로젝트의 애플리케이션 코드가 들어있는 웹서버를 생성하고 이에 대한 테스트를 실행할 수 있도록 하는 테스트 러너
 - Chai : Mocha와 함께 사용할 수 있는 어설션(assertion) 라이브러리
- Jest
 - Facebook에서 만든 테스트 라이브러리
 - 최근에 더 많이 이용됨

Mocha 플러그인을 활용한 테스트



❖ Mocha 플러그인

- vue create 시에 수동 설정을 선택해서 단위 테스트 관련 기능 추가

```
D:\workspace\vuejs2\ch14_appendix1>vue create test1
```

```
Vue CLI v3.0.0-rc.3
```

```
? Please pick a preset: Manually select features
```

```
? Check the features needed for your project:
```

```
(*) Babel  
( ) TypeScript  
( ) Progressive Web App (PWA) Support  
( ) Router  
( ) Vuex  
( ) CSS Pre-processors  
(*) Linter / Formatter  
>(*) Unit Testing  
( ) E2E Testing
```

```
Vue CLI v3.0.0-rc.3
```

```
? Please pick a preset: Manually select features
```

```
? Check the features needed for your project: Babel, Linter, Unit
```

```
? Pick a linter / formatter config: Basic
```

```
? Pick additional lint features: Lint on save
```

```
? Pick a unit testing solution:
```

```
> Mocha + Chai
```

```
Jest
```

Mocha 플러그인을 활용한 테스트



■ 기능 테스트

■ src/sum.js

```
let sum = (a,b)=> {  
  return a+b;  
}  
  
export default sum;
```

■ test/unit/sum.spec.js

```
import { expect } from 'chai'  
import sum from '@sum';  
  
describe('sum', ()=> {  
  it('add 2+3 equals 5', ()=> {  
    expect(sum(2,3)).to.equals(5);  
  })  
})
```

Mocha 플러그인을 활용한 테스트



❧ 테스트 실행

- yarn test:unit

```
WEBPACK Compiling...
[===== ] 95% (emitting)
DONE Compiled successfully in 2301ms
[=====] 100% (completed)
WEBPACK Compiled successfully in 2301ms
MOCHA Testing...

HelloWorld.vue
  ✓ renders props.msg when passed

sum
  ✓ add 2+3 equals 5

2 passing (28ms)
MOCHA Tests completed successfully
Done in 6.23s.
```

Mocha 플러그인을 활용한 테스트



■ HelloWorld.spec.js

```
import { expect } from 'chai'
import { shallowMount } from '@vue/test-utils'
import HelloWorld from '@components/HelloWorld.vue'

describe('HelloWorld.vue', () => {
  it('renders props.msg when passed', () => {
    const msg = 'new message'
    const wrapper = shallowMount(HelloWorld, {
      propsData: { msg }
    })
    //expect(wrapper.text()).to.include(msg)
    expect(wrapper.find('div.hello h1').text()).to.equal(msg)
  })
})
```

Mocha를 이용한 Vue 컴포넌트 테스트



■ todolistapp에 테스트 기능 추가

- vue add @vue/unit-mocha
- test/unit/List.spec.js 작성

```
import { expect } from 'chai'
import { mount, createLocalVue } from '@vue/test-utils'
import Vuex from 'vuex'
import Vue from 'vue'
import store from '@store'
import List from '@components/List.vue'

const localVue = createLocalVue()
localVue.use(Vuex)

describe('List.vue', () => {
  it('초기 렌더링 화면 테스트', (done) => {
    const wrapper = mount(List, { store, localVue });
    Vue.nextTick()
    .then(() => {
      expect(wrapper.vm.$el.querySelectorAll('li').length).to.equal(4)
      done()
    })
    .catch(done)
  })
})
```

Mocha를 이용한 Vue 컴포넌트 테스트



- 테스트 실행(yarn test:unit)

WEBPACK Compiled successfully in 3104ms

MOCHA Testing...

List.vue

✓ 초기 렌더링 화면 테스트

1 passing (26ms)

MOCHA Tests completed successfully

Done in 7.04s.

Mocha를 이용한 Vue 컴포넌트 테스트



■ Vuex의 상태 변화 후 렌더링된 결과 테스트

■ test/unit/List.spec.js 변경

```
import { expect } from 'chai'
import { mount, createLocalVue } from '@vue/test-utils'
import Vuex from 'vuex'
import Vue from 'vue'
import store from '@store'
import List from '@components/List.vue'
import Constant from '@Constant'

const localVue = createLocalVue()
localVue.use(Vuex)

describe('List.vue', () => {
  let wrapper
  before(() => {
    wrapper = mount(List, { store, localVue })
  })
  it('초기 렌더링 화면 테스트', (done) => {
    Vue.nextTick()
    .then(() => {
      expect(wrapper.vm.$el.querySelectorAll('li').length).to.equal(4)
      done()
    })
    .catch(done)
  })
})
```


Mocha를 이용한 Vue 컴포넌트 테스트



■ test/unit/List.spec.js 변경(이어서)

```
it('새로운 todo 추가후 목록 확인', (done) => {
  wrapper.vm.$store.dispatch(Constant.ADD_TODO, { todo:'스쿼시'})
  wrapper.vm.$store.dispatch(Constant.ADD_TODO, { todo:'수영'})
  Vue.nextTick()
  .then(() => {
    var list = wrapper.vm.$el.querySelectorAll('li');
    expect(list[list.length-1].textContent).to.contain('수영')
    expect(list[list.length-2].textContent).to.contain('스쿼시')
    expect(list.length).to.equal(6)
    done()
  })
  .catch(done)
})
})
```

Mocha를 이용한 Vue 컴포넌트 테스트



- 테스트 실행(yarn test:unit)

WEBPACK Compiled successfully in 85712ms

MOCHA Testing...

List.vue

- ✓ 초기 렌더링 화면 테스트
- ✓ 새로운 todo 추가후 목록 확인

2 passing (34ms)

MOCHA Tests completed successfully

Done in 89.68s.

이벤트 시뮬레이션



▣ 브라우저 상에서 일어나는 이벤트를 시뮬레이션

▪ test/unit/List.spec.js 변경

```
import { expect } from 'chai'
import { mount, createLocalVue } from '@vue/test-utils'
import Vuex from 'vuex'
import Vue from 'vue'
import store from '@store'
import TodoList from '@components/TodoList.vue'

const localVue = createLocalVue()
localVue.use(Vuex)

describe('TodoList.vue', () => {
  let wrapper;
  before(() => {
    wrapper = mount(TodoList, { store, localVue });
  })

  it('TodoList 초기 렌더링 테스트', (done) => {
    Vue.nextTick()
    .then(() => {
      expect(wrapper.vm.$el.querySelectorAll('li').length).to.equal(4)
      done()
    })
    .catch(done)
  })
})
```

이벤트 시뮬레이션



■ test/unit/List.spec.js 변경 (이어서)

```
it('클릭이벤트 후 렌더링 결과 확인', (done) => {  
  //입력값을 TodoList의 첫번째 자식컴포넌트인  
  //InputTodo 컴포넌트의 로컬 데이터에 새로운 todo 입력  
  wrapper.vm.$children[0].$data.todo = "피겨강습";  
  
  const evtClick = new window.Event('click');  
  var addbutton = wrapper.vm.$el.querySelector('span.addbutton');  
  addbutton.dispatchEvent(evtClick);  
  wrapper.vm._watcher.run();  
  
  Vue.nextTick()  
    .then(() => {  
      var list = wrapper.vm.$el.querySelectorAll('li');  
      expect(list[list.length-1].textContent).toContain('피겨강습')  
      expect(list.length).toEqual(5)  
      done()  
    })  
    .catch(done)  
  })  
})
```

Jest를 이용한 테스트



■ Jest 개요

- 쉬운 설정 : 설정 파일의 작성을 최소화
- 스냅샷 테스트 : DOM 요소의 스냅샷을 캡처해서 저장하고 다음번 테스트 시에 기존 스냅샷과 비교
- React를 위해 개발되었지만 Vue에서도 지원

■ Jest 설정

- `vue add @vue/unit-jest`
- `jest.config.js` 변경 (필수는 아님)

```
module.exports = {  
  .....(생략)  
  testMatch: [  
    '**/tests/unit/**/*.spec.(js|jsx|ts|tsx)',  
    '**/__tests__/**/*.js|jsx|ts|tsx'  
  ]  
}
```

Jest를 이용한 테스트



■ 간단한 테스트 코드

■ src/sum.js 작성

```
const sum = function(a,b) {  
  return a+b;  
}  
export default sum;
```

■ test/unit/sum.spec.js 작성

```
import sum from '@/sum.js';  
  
describe('sum 테스트', function() {  
  it('1 + 2 = 3', () => {  
    expect(sum(1,2)).toBe(3);  
  })  
})
```

```
yarn run v1.7.0  
$ vue-cli-service test:unit  
PASS tests/unit/sum.spec.js  
sum 테스트  
✓ 1 + 2 = 3 (3ms)
```

```
Test Suites: 1 passed, 1 total  
Tests: 1 passed, 1 total  
Snapshots: 0 total  
Time: 2.312s  
Ran all test suites.  
Done in 4.07s.
```

Jest를 이용한 컴포넌트 테스트



■ 스냅샷 테스트

■ test/unit/List.spec.js

```
import Vue from 'vue'
import Vuex from 'vuex'
import { mount, createLocalVue } from '@vue/test-utils'
import store from '@store'
import List from '@components/List'
import Constant from '@/Constant'

const localVue = createLocalVue()
localVue.use(Vuex)

describe('List.vue', () => {
  let wrapper

  beforeAll(() => {
    wrapper = mount(List, { store, localVue });
    wrapper.vm.$store.dispatch(Constant.ADD_TODO, { todo:'스쿼시'})
  })
```

Jest를 이용한 컴포넌트 테스트



■ test/unit/List.spec.js (이어서)

```
it('초기 렌더링 화면 테스트(Jest)', () => {
  expect(wrapper.vm.$el.querySelectorAll('li').length).toEqual(5);
});

it('스냅샷 테스트(Jest)', (done)=> {
  Vue.nextTick()
    .then(() => {
      expect(wrapper.vm.$el).toMatchSnapshot();
      done()
    })
    .catch(done);
});
```


Jest를 이용한 컴포넌트 테스트



■ 스냅샷 테스트

■ 스냅샷 비교 일치시의 화면

```
yarn run v1.7.0
$ vue-cli-service test:unit
PASS tests/unit/List.spec.js
PASS tests/unit/sum.spec.js
```

```
Test Suites: 2 passed, 2 total
Tests:       3 passed, 3 total
Snapshots:  1 passed, 1 total
Time:        3.054s
Ran all test suites.
Done in 4.75s.
```

■ 스냅샷 비교 불일치시의 화면

```
$ vue-cli-service test:unit
FAIL tests/unit/List.spec.js
  ● List.vue > 스냅샷 테스트(Jest)

    expect(value).toMatchSnapshot()

    Received value does not match stored snapshot "List.vue 스냅샷 테스트(Jest) 1".
```

```
- Snapshot
+ Received

@@ -66,11 +66,11 @@
  </li>
  <li
    class=""
  >
    <span>
      스크시
    </span>
  <!-->
  <span>
```

```
24 |         Vue.nextTick()
25 |         .then(() => {
> 26 |             expect(wrapper.vm.$el).toMatchSnapshot();
    |                                     ^
27 |         })
28 |     })
29 |     .catch(done);
```

at tests/unit/List.spec.js:26:36

```
> 1 snapshot failed.
PASS tests/unit/sum.spec.js
```

Snapshot Summary

> 1 snapshot failed from 1 test suite. Inspect your code changes or re-run jest with `^-u` them.

```
Test Suites: 1 failed, 1 passed, 2 total
Tests:       1 failed, 2 passed, 3 total
Snapshots:   1 failed, 1 total
Time:        3.45s
Ran all test suites.
jest exited with code 1.
```

Jest를 이용한 컴포넌트 테스트



❧ jest 직접 실행

- `npm install -g npx`
- `npx jest` (프로젝트 디렉터리에서)

❧ 스냅샷을 갱신하고 싶다면...

- `npx jest -u`

```
PASS tests/unit/List.spec.js
  > 1 snapshot updated.
PASS tests/unit/sum.spec.js
```

```
Snapshot Summary
  > 1 snapshot updated from 1 test suite.
```

```
Test Suites: 2 passed, 2 total
Tests:       3 passed, 3 total
Snapshots:   1 updated, 1 total
Time:        2.742s, estimated 3s
Ran all test suites.
```