

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

- webpack : 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 어썰션(assertion) 라이브러리
 - 최근 버전에 Jest 추가. Jest VS Mocha+Karma 조합중 선택 가능

webpack 템플릿을 이용한 테스트



■ 간단한 테스트용 프로젝트 생성

■ vue init webpack test01

```
D:\workspace\vuejs>vue init webpack test01
```

```
? Project name test01
? Project description A Vue.js project
? Author stepanowon <stepanowon@hotmail.com>
? Vue build standalone
? Install vue-router? Yes
? Use ESLint to lint your code? No
? Setup unit tests Yes
? Pick a test runner karma
? Setup e2e tests with Nightwatch? No
```

```
vue-cli · Generated "test01".
```

```
To get started:
```

```
cd test01
npm install
npm run dev
```

```
Documentation can be found at https://vuejs-templates.github.io/webpack
```

- 단위 테스트 수행 여부를 Yes로 지정하고 테스트 러너를 karma로 지정
- 버전업된 부분

■ cd test01

■ npm install

webpack 템플릿을 이용한 테스트



■ 기본 기능 테스트

- 예제 A-01 : src/sum.js

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

- 예제 A-02 : test/unit/specs/sum.spec.js

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

- npm run test
 - karma 테스트 러너를 이용함

webpack 템플릿을 이용한 테스트



■ 실행 결과

```
D:\workspace#\vuejs#\test01>npm run test
```

```
> test01@1.0.0 test D:\workspace#\vuejs#\test01
> npm run unit
```

```
> test01@1.0.0 unit D:\workspace#\vuejs#\test01
> cross-env BABEL_ENV=test karma start test/unit/karma.conf.js --single-run
```

```
14 11 2017 08:52:08.705: INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
14 11 2017 08:52:08.721: INFO [launcher]: Launching browser PhantomJS with unlimited concurrency
14 11 2017 08:52:08.736: INFO [launcher]: Starting browser PhantomJS
14 11 2017 08:52:11.761: INFO [PhantomJS 2.1.1 (Windows 8 0.0.0)]: Connected on socket YQnB5jzjCxeTU3FhAAAA with id 90194019
```

```
HelloWorld.vue
✓ should render correct contents
```

```
PhantomJS 2.1.1 (Windows 8 0.0.0): Executed 1 of 1 SUCCESS (0.015 secs / 0.01 secs)
TOTAL: 1 SUCCESS
```

```
===== Coverage summary =====
Statements : 100% ( 2/2 )
Branches   : 100% ( 0/0 )
Functions  : 100% ( 0/0 )
Lines      : 100% ( 2/2 )
=====
```

webpack 템플릿을 이용한 테스트



- 예제 A-04 : test/unit/specs/HelloWorld.spec.js
 - 업그레이드된 Vue-CLI에서는 HelloWorld.vue를 테스트함.

```
import Vue from 'vue'
import HelloWorld from '@components/HelloWorld'

describe('HelloWorld.vue', () => {
  it('should render correct contents', () => {
    const Constructor = Vue.extend(HelloWorld)
    const vm = new Constructor().$mount()
    expect(vm.$el.querySelector('.hello h1').textContent)
      .to.equal('Welcome to Your Vue.js App')
  })
})
```

- .to.equal() --> Matcher
 - Chai가 지원하는 함수
 - 다양한 Matcher를 알아보려면 <http://chaijs.com> 확인

webpack 템플릿을 이용한 테스트



■ Vue 컴포넌트 테스트

- 13장의 todolistapp을 이용해 테스트
 - webpack-simple 템플릿은 테스트 기능을 포함하고 있지 않기 때문에 webpack 템플릿으로 새로운 프로젝트를 생성한 후 코드를 옮기는 방법 사용
- `vue init webpack todolistapp_test1`

```
D:\workspace\vuejs>vue init webpack todolistapp_test1
```

```
? Project name todolistapp_test1
? Project description A Vue.js project
? Author stepanowon <stepanowon@hotmail.com>
? Vue build standalone
? Install vue-router? Yes
? Use ESLint to lint your code? No
? Setup unit tests Yes
? Pick a test runner karma
? Setup e2e tests with Nightwatch? No
```

```
vue-cli · Generated "todolistapp_test1".
```

To get started:

```
cd todolistapp_test1
npm install
npm run dev
```

Documentation can be found at <https://vuejs-templates.github.io/webpack>

webpack 템플릿을 이용한 테스트



- 프로젝트 생성 후...
 - 기존 src 디렉토리를 삭제
 - 13장에서 완성된 todolistapp의 src 디렉토리를 복사함.
 - npm run dev로 실행 --> 실행 여부 확인
- 테스트해 볼 컴포넌트 : List.vue
 - Vuex의 상태를 이용해 할일 목록을 출력함.
 - 따라서 컴포넌트 인스턴스에 store를 주입한 후 테스트가 수행되어야 함.

webpack 템플릿을 이용한 테스트



■ 예제 A-05 : test/unit/specs/List.spec.js

```
import Vue from 'vue'
import Vuex from 'vuex'
import store from '@./store'
import List from '@./components/List.vue'
import constant from '@./constant'

describe('List.vue', () => {
  it('초기 렌더링 화면 테스트', (done) => {
    const vm = new Vue({
      template: '<div><list></list></div>',
      store,
      components: {
        'list': List
      }
    }).$mount()

    Vue.nextTick()
      .then(() => {
        expect(vm.$el.querySelectorAll('li').length).to.equal(4)
        done()
      })
      .catch(done)
  })
})
```


webpack 템플릿을 이용한 테스트



■ 예제 A-06 : src/store/index.js

```
import Vue from 'vue';
import Vuex from 'vuex';
import state from './state.js';
import mutations from './mutations.js';
import actions from './actions.js';
import es6Promise from 'es6-promise';
es6Promise.polyfill();
```

```
Vue.use(Vuex);
```

```
const store = new Vuex.Store({
  state,
  mutations,
  actions
})
```

```
export default store;
```

webpack 템플릿을 이용한 테스트



■ 컴포넌트 테스트 수행 결과

```
D:\workspace\vuejs\todolistapp_test1>npm run unit
```

```
> todolistapp_test1@1.0.0 unit D:\workspace\vuejs\todolistapp_test1
> cross-env BABEL_ENV=test karma start test/unit/karma.conf.js --single-run
```

```
14 11 2017 09:17:33.512: INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
14 11 2017 09:17:33.528: INFO [launcher]: Launching browser PhantomJS with unlimited concurrency
14 11 2017 09:17:33.544: INFO [launcher]: Starting browser PhantomJS
14 11 2017 09:17:36.629: INFO [PhantomJS 2.1.1 (Windows 8 0.0.0)]: Connected on socket CiRBsfySPxy1uvHzAAAA with id 37798
558
```

```
List.vue
```

```
✓ 초기 렌더링 화면 테스트
```

```
PhantomJS 2.1.1 (Windows 8 0.0.0): Executed 1 of 1 SUCCESS (0.047 secs / 0.019 secs)
TOTAL: 1 SUCCESS
```

```
===== Coverage summary =====
Statements : 30% ( 6/20 )
Branches   : 50% ( 2/4 )
Functions  : 11.11% ( 1/9 )
Lines      : 26.32% ( 5/19 )
=====
```

webpack 템플릿을 이용한 테스트



- Vuex Store의 액션을 dispatch 하여 상태를 변경
 - 상태 변경으로 인한 렌더링된 요소의 결과를 비교
- 예제 A-07

```
it('새로운 todo 추가후 목록 확인', (done) => {
  const vm = new Vue({
    template: '<div><list></list></div>',
    store,
    components: {
      'list': List
    }
  }).$mount()

  vm.$store.dispatch(constant.ADD_TODO, { todo:'스쿼시'})
  vm.$store.dispatch(constant.ADD_TODO, { todo:'수영'})

  Vue.nextTick()
    .then(() => {
      var list = vm.$el.querySelectorAll('li');
      expect(list[list.length-1].textContent).toContain('수영')
      expect(list[list.length-2].textContent).toContain('스쿼시')
      expect(list.length).toEqual(6)
      done()
    })
    .catch(done)
})
```

webpack 템플릿을 이용한 테스트



■ 이벤트 시뮬레이션

- 브라우저 화면상에서 발생하는 이벤트를 시뮬레이션
 - 내부적으로 PhantomJS 를 이용함
- PhantomJS : 헤드리스 웹 브라우저
 - GUI(Graphic User Interface)가 없는 웹브라우저
 - 테스트 프레임워크에서 테스트할 때 사용
 - API로 웹 페이지를 직접 조작할 수 있음
- 예제 A-08

```
const vm = new Vue({
  template: '<div><input-todo ref="a"></input-todo><list></list></div>',
  store,
  components: {
    'list': List,
    'input-todo' : InputTodo,
  }
}).$mount()
```

webpack 템플릿을 이용한 테스트



■ 예제 A-08

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

  it('새로운 todo 추가후 목록 확인', (done) => {
    vm.$refs.a._data.todo = "수영강습";
    const evtClick = new window.Event('click');
    var addbutton = vm.$el.querySelector('span.addbutton');
    addbutton.dispatchEvent(evtClick);
    vm._watcher.run();

    Vue.nextTick()
      .then(() => {
        var list = vm.$el.querySelectorAll('li');
        expect(list[list.length-1].textContent).to.contain('수영강습')
        expect(list.length).to.equal(5)
        done()
      })
      .catch(done)
  })
})
```

webpack 템플릿을 이용한 테스트



■ 테스트 결과

```
> todolistapp_test1@1.0.0 unit D:\workspace\vuejs\ch14_appendix1\todolistapp_test1
> cross-env BABEL_ENV=test karma start test/unit/karma.conf.js --single-run

25 07 2017 19:33:57.818: INFO [karma]: Karma v1.7.0 server started at http://0.0.0.0:9876/
25 07 2017 19:33:57.818: INFO [launcher]: Launching browser PhantomJS with unlimited concurrency
25 07 2017 19:33:57.914: INFO [launcher]: Starting browser PhantomJS
25 07 2017 19:34:01.050: INFO [PhantomJS 2.1.1 (Windows 8 0.0.0)]: Connected on socket RcqBfu9bgfk6Yuz9AAAA with id 15000944

List.vue
  ✓ 초기 렌더링 화면 테스트
  ✓ 새로운 todo 추가후 목록 확인

PhantomJS 2.1.1 (Windows 8 0.0.0): Executed 2 of 2 SUCCESS (0.03 secs / 0.013 secs)
TOTAL: 2 SUCCESS

===== Coverage summary =====
Statements : 75% ( 12/16 )
Branches   : 75% ( 3/4 )
Functions  : 55.56% ( 5/9 )
Lines      : 73.33% ( 11/15 )
=====
```

Jest를 이용한 테스트



❑ Vue-CLI 이전 버전은 Karma+Mocha만 지원

- $\leq 2.8.X$ 까지
- Vue-CLI 2.9.x 버전 부터는 Jest가 지원되고 Default 설정

❑ 책은 2.7.x 버전

- webpack-simple 프로젝트 템플릿에 직접 관련 라이브러리를 추가함.
 - `npm install --save-dev babel babel-jest jest jest-vue-preprocessor`

❑ 이 강의 자료에서는 업그레이드된 webpack 템플릿을 이용함

Jest를 이용한 테스트



■ 프로젝트 생성

- `vue init webpack todolistapp_test2`
- `cd todolistapp_test2`
- `npm install`
- `npm install --save vuex es6-promise`
- `package.json` 확인
 - `jest`, `vue-jest`가 이미 포함되어 있음
- 생성한 프로젝트의 `src` 디렉터리 삭제 후 13장 `todolistapp`의 `src` 디렉터리를 복사
- `npm run dev`로 실행 여부 확인

Jest를 이용한 테스트



■ 새로이 생성된 프로젝트의 테스트 케이스 작성 디렉터리

- test/unit/specs/*
- HelloWorld.spec.js 파일은 삭제

■ 간단한 테스트

- 예제 A-11 : src/sum.js

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

- 예제 A-12 : test/unit/specs/sum.spec.js

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

Jest를 이용한 테스트



- 테스트!!
 - jest --coverage

```
PS D:\workspace\vuejs\todolistapp_test2> jest --coverage
```

```
PASS test\unit\specs\sum.spec.js
```

```
sum 테스트
```

```
✓ 1 + 2 = 3
```

```
Test Suites: 1 passed, 1 total
```

```
Tests: 1 passed, 1 total
```

```
Snapshots: 0 total
```

```
Time: 0.751s, estimated 1s
```

```
Ran all test suites.
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
All files	0.72	0	0	1.39	
src	33.33	100	100	50	
constant.js	0	100	100	0	1
sum.js	100	100	100	100	
src/components	0	0	0	0	
HelloWorld.vue	0	0	0	0	... 39,40,41,42
InputTodo.vue	0	0	0	0	... 46,47,48,49
List.vue	0	0	0	0	... 90,91,92,93
TodoList.vue	0	0	0	0	... 43,44,45,46
src/store	0	0	0	0	
actions.js	0	0	0	0	... 10,11,14,15
index.js	0	0	0	0	... ,4,5,7,9,12
mutations.js	0	0	0	0	1,5,6,10,13
state.js	0	0	0	0	1

Jest를 이용한 테스트



■ Vue 컴포넌트에 대한 테스트

■ 스냅샷 테스트

- 한 시점의 스냅샷을 저장해두었다가 코드가 변경된 경우에 새로운 스냅샷을 만들어 비교하는 테스트 방법
- 예상치 못한 UI의 변경을 확인할 수 있는 유용한 방법

■ 잠시 트랜지션 효과 기능은 off 시키자.

- 비동기로 서서히 UI가 변화하는 경우 스냅샷 테스트에 어려움이 있음
- 예제 A-13 : src/components/List.vue

```
<template>
  <!--<transition-group name="list" tag="ul">-->
  <ul>
    <li v-for="(a, index) in todolist" v-bind:class="checked(a.done)"
      v-on:click="doneToggle({index:index})" :key="index">
      <span>{{ a.todo }}</span>
      <span v-if="a.done"> (완료)</span>
      <span class="close" v-on:click.stop="deleteTodo({index:index})">✕</span>
    </li>
  </ul>
  <!--</transition-group>-->
</template>
```

Jest를 이용한 테스트



■ 예제 A-14 : test/unit/specs/List.spec.js

```
import Vue from 'vue'
import Vuex from 'vuex'
import store from '../../src/store'
import List from '../../src/components/List'
import constant from '../../src/constant'

describe('List.vue', () => {
  var vm;
  beforeEach(() => {
    vm = new Vue({
      store,
      render: h => h(List)
    }).$mount()
    vm.$store.dispatch(constant.ADD_TODO, { todo:'스쿼시'})
  })
  it('초기 렌더링 화면 테스트', () => {
    expect(vm.$el.querySelectorAll('li').length).toEqual(5);
  });
  it('스냅샷 테스트', (done)=> {
    Vue.nextTick()
    .then(() => {
      expect(vm.$el).toMatchSnapshot();
      done()
    })
    .catch(done);
  })
});
```

Jest를 이용한 테스트



■ jest --coverage

[스냅샷 생성]

```
PS D:\workspace\vuejs\todolistapp_test2> jest --coverage
PASS test\unit\specs\List.spec.js
  ● Console
```

```
    console.log src\store\actions.js:5
      ###addTodo!!!
    console.log src\store\actions.js:6
      { todolist: [Getter/Setter] }
```

```
> 1 snapshot written.
```

```
PASS test\unit\specs\sum.spec.js
```

Snapshot Summary

```
> 1 snapshot written in 1 test suite.
```

```
Test Suites: 2 passed, 2 total
```

```
Tests: 3 passed, 3 total
```

```
Snapshots: 1 added, 1 total
```

```
Time: 1.263s
```

```
Ran all test suites.
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
All files	8.75	2.38	0	13.46	
src	100	100	100	100	
sum.js	100	100	100	100	
src/components	0	0	0	0	
InputTodo.vue	0	0	0	0	... 46,47,48,49
List.vue	0	0	0	0	... 90,91,92,93
TodoList.vue	0	0	0	0	... 43,44,45,46
src/store	50	50	100	50	
actions.js	42.86	100	100	42.86	10,11,14,15
index.js	100	100	100	100	
mutations.js	50	50	100	50	10,13

[스냅샷이 일치하지 않을때]

```
PS D:\workspace\vuejs\todolistapp_test2> jest --coverage
FAIL test\unit\specs\List.spec.js
  ● Console
```

```
    console.log src\store\actions.js:5
      ###addTodo!!!
    console.log src\store\actions.js:6
      { todolist: [Getter/Setter] }
```

● List.vue > 스냅샷 테스트

```
expect(value).toMatchSnapshot()
```

```
Received value does not match stored snapshot 1.
```

```
- Snapshot
```

```
+ Received
```

```
@@ -63,11 +63,11 @@
```

```
</li>
```

```
<li
```

```
  class=""
```

```
>
```

```
<span>
```

```
-   스쿼시
```

```
+   테니스
```

```
</span>
```

```
<!-->
```

```
<span
```

```
</span>
```

```
at test/unit/specs/List.spec.js:25:28
```

```
at <anonymous>
```

```
at process._tickCallback (internal/process/next_tick.js:188:7)
```

```
> 1 snapshot test failed.
```

Jest를 이용한 테스트



- 저장된 스냅샷은 __snapshots__ 디렉터리에 저장됨.
- 스냅샷을 새로운 버전으로 갱신하고 싶다면
 - `jest --updateSnapshot`