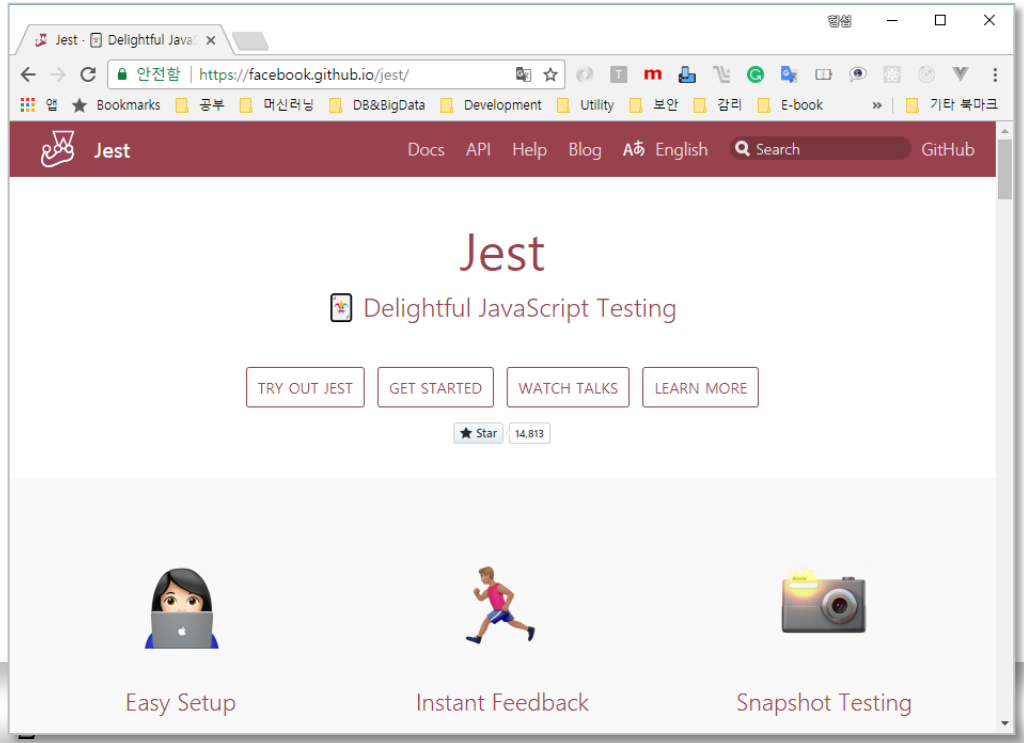
1. 테스트 개요

- **새로운 기능, 컴포넌트 추가시마다 버그 발생 여부 확인 필요**
  - 반드시 필요하다고 볼수는 없지만 때로는 유용함.
- **자동화된 테스트**
  - 예상되는 동작을 동적으로 문서화하고
  - 문제 발생시 즉시 발견할 수 있도록 함

2. Jest 개요

Jest란?

- Delightful Javascript Testing!!
- React를 비롯해 모든 JS 코드를 테스트하기 위해 페이스북에서 사용되는 테스팅 프레임워크
  - 가상 DOM 기반으로 테스트를 수행함
  - JSX을 기본적으로 지원함.
  - jasmine기반으로 작성되었음
- 지원
  - react, react native
  - Async
  - webpack
  - jquery



3. Jest 기능 테스트(1)

테스트 프로젝트 생성

- mkdir jesttest
- yarn init
- yarn add -D jest-cli babel-jest babel-preset-env
- .babelrc 파일 작성

```
{
  "presets" : [ "env" ]
}
```

- package.json 에서 scripts 수정

```
"scripts": {
   "test": "jest"
}
```

- 테스트할 함수 작성(sum.js)
- __tests__ 디렉토리에 sum-test.js 파일 작성

3. Jest 기능 테스트(2)

테스트 실행

- yarn test

```
yarn test v0.27.5
$ jest
 PASS  __tests__\sum-test.js
  sum
    √ adds 1+2 equal 3 (3ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        4.082s
Ran all test suites.
Done in 5.21s.
```

3. Jest 기능 테스트(3)

Global API

- test(name, fn), test.only(name, fn), test.skip(name,fn)
  - Alias : it(...), it.onley(...), it.skip(...)
- afterAll(fn), afterEach(fn)
- beforeAll(fn), beforeEach(fn)
- describe(name, fn), describe.only(name, fn), describe.skip(name,fn)
- require.requireActual(moduleName), require.requireMock(moduleName)

3. Jest 기능 테스트(4)

- Matcher?
  - Jest에서 값이 같은지 비교하기 위해 사용하는 함수
- 다양한 Matcher
  - toBe() : 같은지 비교 (===)
  - toEqual() : 객체의 값이 같은지를 비교할 때는 toBe 대신 toEqual 사용
  - not : toBe()의 반대를 사용하고 싶을 때
    - expect(a+ b).not.toBe(0)
  - 기타
    - toBeNull(), toBeUndefined(), toBeDefined(), toBeTruthy(), toBeFalsy()
    - toBeGreaterThan(), toBeGreaterThanOrEqual(), ...
    - toMatch() : 정규식을 이용한 매칭 여부 조사
    - toContain() : 배열값에 특정값이 포함되어있는지 여부 조사

4. Snapshot 테스트(1)

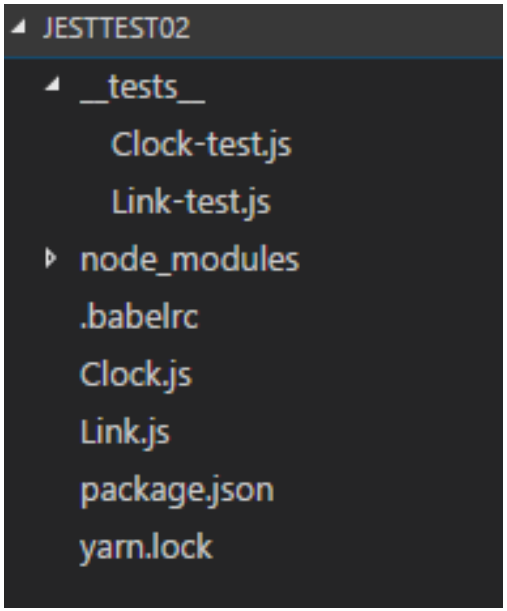
Snapshot 테스트?

- Snapshot?
  - 렌더링된 컴포넌트의 출력 결과 UI
  - 렌더러를 이용해 React 트리를 직렬화한 문자열 값
- Snapshot 테스트?
  - UI가 예기치 않게 변경되지 않았는지 여부를 확인하기 위해 수행하는 테스트 방법 중의 하나
- jest에서 Snapshot 테스트를 위해 react-test-renderer를 사용함
  - yarn add -D react-test-renderer
- 한번 렌더링된 결과는 __snapshots__ 디렉토리에 저장됨
  - 다음 번 테스트 수행시 렌더링된 결과와 snapshot을 비교하게 됨.

4. Snapshot 테스트(2)

- package.json, .babelrc

```
{
  "name": "jesttest",
  "version": "1.0.0",
  "main": "index.js",
  "license": "MIT",
  "scripts": {
    "test": "jest"
  },
  "devDependencies": {
    "babel-jest": "^22.0.6",
    "babel-preset-env": "^1.6.1",
    "babel-preset-react": "^6.24.1",
    "jest-cli": "^22.0.6",
    "react-test-renderer": "^16.2.0"
  },
  "dependencies": {
    "react": "^16.2.0"
  }
}
```




```
{
   "presets" : ["env", "react"]
}
```

4. Snapshot 테스트(3)

Link.js

```
import React from 'react';

const STATUS = {
  HOVERED: 'hovered',
  NORMAL: 'normal',
};

class Link extends React.Component {

  constructor() {
    super();

    this._onMouseEnter = this._onMouseEnter.bind(this);
    this._onMouseLeave = this._onMouseLeave.bind(this);

    this.state = {
      class: STATUS.NORMAL,
    };
  }
```

```
  _onMouseEnter() {
    this.setState({class: STATUS.HOVERED});
  }

  _onMouseLeave() {
    this.setState({class: STATUS.NORMAL});
  }

  render() {
    return (
      <a
        className={this.state.class}
        href={this.props.page || '#'}
        onMouseEnter={this._onMouseEnter}
        onMouseLeave={this._onMouseLeave}>
        {this.props.children}
      </a>
    );
  }
}

export default Link;
```

4. Snapshot 테스트(4)

Clock.js

```
import React from 'react';

class Clock extends React.Component {

  constructor() {
    super();
    this.state = {seconds: Date.now() / 1000};
  }

  componentDidMount() {
    this.timerID = setInterval(
      () => this.tick(),
      1000
    );
  }

  componentWillUnmount() {
    clearInterval(this.timerID);
  }

  tick() {
    this.setState({
      seconds: Date.now() / 1000,
    });
  }

  render() {
    return (
      <p>{this.state.seconds} seconds
           have ellapsed since the UNIX epoch.</p>
    );
  }
}

export default Clock;
```

4. Snapshot 테스트(5)

- Clock-test.js

```
'use strict';

import React from 'react';
import Clock from '../Clock';
import renderer from 'react-test-renderer';

jest.useFakeTimers();
Date.now = jest.fn(() => 1482363367071);

it('renders correctly', () => {
  const tree = renderer.create(
    <Clock />
  ).toJSON();
  expect(tree).toMatchSnapshot();
});
```

4. Snapshot 테스트(6)

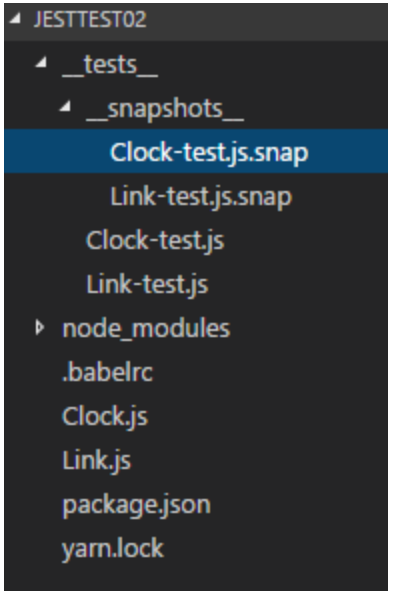
Link-test.js

```
'use strict';

import React from 'react';
import Link from '../Link';
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();
});

it('renders as an anchor when no page is set', () => {
  const tree = renderer.create(
    <Link>Facebook</Link>
  ).toJSON();
  expect(tree).toMatchSnapshot();
});

it('properly escapes quotes', () => {
  const tree = renderer.create(
    <Link>{'"Facebook" ₩₩₩'is ₩₩ ₩'awesome₩''}</Link>
  ).toJSON();
  expect(tree).toMatchSnapshot();
});

it('changes the class when hovered', () => {
  const component = renderer.create(
    <Link
page="http://www.facebook.com">Facebook</Link>
  );
  let tree = component.toJSON();
  expect(tree).toMatchSnapshot();

  // manually trigger the callback
  tree.props.onMouseEnter();
  // re-rendering
  tree = component.toJSON();
  expect(tree).toMatchSnapshot();

  // manually trigger the callback
  tree.props.onMouseLeave();
  // re-rendering
  tree = component.toJSON();
  expect(tree).toMatchSnapshot();
});
```

4. Snapshot 테스트(7)

yarn test

- 처음 수행할 경우 __snapshots__ 디렉토리에 문자열 생성
- 다음 번 수행시 저장된 문자열과 비교하여 다르면 테스트가 실패함.



시작 Clock-test.js.snap ×

```
1  // Jest Snapshot v1, https://goo.gl/fbAQLP
2
3  exports[`renders correctly 1`] = `
4  <p>
5    1482363367.071
6    seconds have ellapsed since the UNIX epoch.
7  </p>
8  `;
```

5. DOM 테스트(1)

DOM 테스트?

- React Component가 렌더링한 결과를 테스트함.
- enzyme과 react-addons-test-utils 사용
  - yarn add -D enzyme enzyme-adapter-react-16 react-addons-test-utils
- enzyme
  - airbnb에서 만든 React 컴포넌트의 산출물을 탐색, 조작, 확인 주장(Assertion)을 쉽게 하는 테스팅 유틸리티

Shallow Rendering

- 종속된 자식 컴포넌트는 렌더링하지 않음
  - 해당 컴포넌트의 렌더링 결과만 테스트할 수 있다는 장점이 있음.
  - 자식 컴포넌트의 오류로 인한 테스트 중단을 예방할 수 있음.
- rendering 함수
  - shallow(), mount(), render()
  - 각 렌더링 함수의 리턴 값 객체를 이용해 사용할 수 있는 다양한 메서드 제공

5. DOM 테스트(2)

테스트를 위한 환경 생성

- 의존성 설정
  - yarn add react react-dom
  - yarn add -D babel-jest babel-preset-env babel-preset-react enzyme enzyme-adapter-react-16 jest react-addons-test-utils
- .babelrc 설정
  - 이전과 동일하게...

5. DOM 테스트(3)

Checkbox.js

- 간단한 컴포넌트

```
import React from 'react';

class Checkbox extends React.Component {

  constructor(props) {
    super(props);
    this.state = {isChecked: false};

    this.onChange = this.onChange.bind(this);
  }

  onChange() {
    this.setState({isChecked: !this.state.isChecked});
  }
```

```
 render() {
    return (
      <label>
        <input
          type="checkbox"
          checked={this.state.isChecked}
          onChange={this.onChange}
        />
        {this.state.isChecked ? this.props.labelOn :
this.props.labelOff}
      </label>
    );
  }
}

export default Checkbox;
```

5. DOM 테스트(4)

__tests__/Checkbox-test.js

```
import React from 'react';
import Checkbox from '../Checkbox';

import Enzyme, {shallow} from 'enzyme';
import Adapter from 'enzyme-adapter-react-16';
Enzyme.configure({ adapter: new Adapter() });


test('Checkbox changes the text after click', () => {
    const checkbox = shallow(
      <Checkbox labelOn="On" labelOff="Off" />
    );

    expect(checkbox.text()).toEqual('Off');
     //이벤트 시뮬레이션
    checkbox.find('input').simulate('change');

    expect(checkbox.text()).toEqual('On');
});
```

5. DOM 테스트(5)

테스트 수행

- yarn test

```
yarn test v0.27.5
$ jest
 PASS  __tests__\Checkbox-test.js
  √ Checkbox changes the text after click (16ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.967s, estimated 2s
Ran all test suites.
Done in 2.58s.
```

6. create-react-app에서의 테스트(1)

- create-react-app는...
  - jest 설정은 이미 되어 있음
    - 스냅샷 테스트를 위한 react-test-renderer는 포함되어 있지 않음.
  - enzyme 관련 패키지는 설치되어 있지 않음
  - 테스트 매치
    - *.test.js, *.spec.js, __tests__/*.js
  - 테스트 결과 watch 기능 작동
    - yarn test
    - 코드를 변경하고 저장하면 자동으로 테스트
- 패키지 참조
  - yarn add -D enzyme enzyme-adapter-react-v16 react-addons-test-utils react-test-renderer

6. create-react-app에서의 테스트(2)

- Clock, Checkbox 컴포넌트로 테스트한 결과

```
PS D:\workspace\react2\ch10\test99> jest --coverage --env=jsdom
 PASS  src\App.test.js
 PASS  src\Checkbox.test.js
 PASS  src\Clock.test.js

Test Suites: 3 passed, 3 total
Tests:       3 passed, 3 total
Snapshots:   1 passed, 1 total
Time:        2.163s
Ran all test suites.
--------------------------|----------|----------|----------|----------|----------------|
File                      |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
--------------------------|----------|----------|----------|----------|----------------|
All files                 |    17.24 |     7.41 |       28 |    29.41 |                |
 App.js                   |      100 |      100 |      100 |      100 |                |
 Checkbox.js              |      100 |      100 |      100 |      100 |                |
 Clock.js                 |    57.14 |      100 |       50 |    57.14 |       12,18,22 |
 index.js                 |        0 |        0 |        0 |        0 |  1,2,3,4,5,7,8 |
 registerServiceWorker.js |        0 |        0 |        0 |        0 |... 126,127,128 |
--------------------------|----------|----------|----------|----------|----------------|
```