Jest

Overview

- From https://facebook.github.io/jest/ ...

- "Built on top of Jasmine"
- "Automatically finds tests to execute in your repo"
  - by default, all `.js` files in `__tests__` directory
- "Automatically mocks CommonJS modules returned by require(), making most existing code testable"
- "DOM APIs are mocked and tests run in parallel via a small Node.js command line utility"
- "Allows you to test asynchronous code synchronously"
- "Runs your tests with a fake DOM implementation (via jsdom) so that your tests can run on the command line"

Can Test React Components

- Example files see https://github.com/mvolkmann/react-examples/tree/master/jest
  - **`webpack.config.js`** - configures webpack to use Babel (ES6 to ES5 transpilation) and ESLint (JavaScript linting) loaders
  - **`.babelrc`** - configures Babel "preset" configurations to be used
  - **`package.json`** - used by npm; specifies dependencies, defines test script, and configures jest
  - **`jest/preprocessor.js`** - required in order to test code that uses JSX
  - **`src/greeting.js`** - defines a React component to be tested
  - **`__tests__/greeting-test.js`** - defines tests
  - **`__tests__/dont-mock.js`** - work-around for import hoisting issue (see comments in test code)
- Steps to run tests
  - **`npm install`**
  - **`npm test`** - runs **`jest`** command

webpack.config.js

- webpack is a JavaScript module bundler
- It follows dependencies expressed with ES 2015 imports, AMD, and CommonJS (require) to produce a single JavaScript file that combines many

```
const path = require('path');
const jsPath = path.join(__dirname, 'src');

module.exports = {
  entry: './src/demo.js',
  output: {
    path: __dirname,
    filename: 'build/bundle.js'
  },
  module: {
    loaders: [
      {test: jsPath, loader: 'babel-loader'},
      {test: jsPath, loader: 'eslint-loader'}
    ]
  },
  resolve: {
    root: '.'
  }
};
```

uses the component being tested

.babelrc

- Babel is a JavaScript transpiler
- It can be configured to
  transpile ES 2015 code to ES5
  and convert JSX to JavaScript functions (React)

```
{
  "presets": ["es2015", "react"]
}
```

package.json

- Used by npm
- Specifies dependencies, defines **`test`** script, and configures Jest

```
{
  "name": "jest-demo",
  "version": "1.0.0",
  "description": "This demonstrates using jest to test React components.",
  "main": "index.html",
  "scripts": {
    "test": "jest"
  },
  "devDependencies": {
    "babel-core": "*",
    "babel-jest": "*",
    "babel-loader": "*",
    "babel-preset-es2015": "*",
    "babel-preset-react": "*",
    "eslint": "*",
    "eslint-loader": "*",
    "eslint-plugin-react": "*",
    "jest-cli": "*",
    "jest-webpack-alias": "*",
    "react-addons-test-utils": "*",
    "webpack": "*",
    "webpack-dev-server": "*"
  },
  "dependencies": {
    "react": "*",
    "react-dom": "*"
  },
  "jest": {
    "scriptPreprocessor": "jest/preprocessor.js",
    "unmockedModulePathPatterns": [
      "node_modules/react",
      "node_modules/react-dom",
      "node_modules/react-addons-test-utils",
      "node_modules/fbjs"
    ]
  }
}
```

don't mock React libraries or the fbjs utility library that React uses

preprocessor.js

- Required in order to test code that uses JSX

```
const babelJest = require('babel-jest');
const webpackAlias = require('jest-webpack-alias');

module.exports = {
  process(src, filename) {
    if (filename.indexOf('node_modules') === -1) {
      src = babelJest.process(src, filename);
      src = webpackAlias.process(src, filename);
    }
    return src;
  }
};
```

processes all JavaScript files except those below the `node_modules` directory

greeting.js

- The React component to be tested

```
import React from 'react';

class Greeting extends React.Component {
  constructor(props) {
    super(props);
    this.state = {name: 'World'}; // initial state
  }

  setName(event) {
    this.setState({name: event.target.value});
  }

  render() {
    return (
      <form>
        <div>
          <label>Name: </label>
          <input type="text" value={this.state.name}
            onChange={e => this.setName(e)}/>
        </div>
        <div>
          {this.props.greet}, {this.state.name}!
        </div>
      </form>
    );
  }
}

Greeting.defaultProps = {greet: 'Hello'};
```

greeting-test.js ...

```
/* global describe, expect, it */

// All ES6 imports get hoisted to the top,
// so they get mocked before the line below runs!
//jest.dontMock('../src/greeting');
// This line was moved to dont-mock.js so it can be imported first.
// See explanation at https://github.com/babel/babel-jest/issues/16.
import './dont-mock';

import React from 'react';
import ReactDOM from 'react-dom';
import TestUtils from 'react-addons-test-utils';

import Greeting from '../src/greeting';

describe('greeting', () => {
  it('greets with defaults', () => {
    const component = TestUtils.renderIntoDocument(
      <Greeting/>
    );
    const form = ReactDOM.findDOMNode(component);
    const lastDiv = form.lastChild;
    expect(lastDiv.textContent).toBe('Hello, World!');
  });
```

This is the part to focus on. Notice how easy it is to write the actual test code!

... greeting-test.js

```
  it('greets with prop', () => {
    const component = TestUtils.renderIntoDocument(
      <Greeting greet="Hola"/>
    );
    const form = ReactDOM.findDOMNode(component);
    const lastDiv = form.lastChild;
    expect(lastDiv.textContent).toBe('Hola, World!');
  });

  it('greets with state', () => {
    const component = TestUtils.renderIntoDocument(
      <Greeting/>
    );
    component.setState({name: 'Mark'});
    const form = ReactDOM.findDOMNode(component);
    const lastDiv = form.lastChild;
    expect(lastDiv.textContent).toBe('Hello, Mark!');
  });
});
```

no-mock.js

- Prevents the code being tested from being mocked

```
/* global jest */
jest.dontMock('../src/greeting');
```

Running Tests

- Enter "npm test"
- Output

```
> jest-demo@1.0.0 test /Users/Mark/Documents/training/React/react-examples/jest
> jest

Using Jest CLI v0.8.0, jasmine1
 PASS  __tests__/dont-mock.js (0.339s)
 PASS  __tests__/greeting-test.js (0.753s)
1 test passed (1 total in 2 test suites, run time 1.668s)
```