

Testing Vue Components

Why tests are important?

- Less debug time
- Code proven to meet requirements
- Documentation!
- Reduce chances of bugs in new / existing features
- Improve design
- Reduce fear

The three laws of TDD

1. You are not allowed to write any production code unless it is to make a failing unit test pass.
2. You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
3. You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

Demo

Install dependencies

```
npm init -y  
npm install vue  
npm install expect jsdom jsdom-global mocha mocha-webp
```

package.json

```
"scripts": {  
  "test": "mocha-webpack --webpack-config webpack.conf  
  "watch": "mocha-webpack --webpack-config webpack.con  
}
```

test/setup.js

```
require('jsdom-global')();
```

webpack.config.js

```
module.exports = {  
  module: {  
    rules: [  
      {  
        test: /\.vue$/,  
        use: 'vue-loader'  
      }  
    ]  
  }  
};
```


src/components/Counter.vue

```
<template>
  <div>
    <span class="count" v-text="count"></span>
    <button @click="count++" class="increment">Increment</button>
    <button @click="count--" class="decrement" v-show="count > 0">Decrement</button>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        count: 0
      };
    }
  }
</script>
```

test/counter.spec.js

```
import { mount } from 'vue-test-utils';
import Counter from '../src/components/Counter.vue';
import expect from 'expect';

describe ('Counter', () => {
  let wrapper;

  beforeEach(() => {
    wrapper = mount(Counter);
  });

  it('defaults to a count of 0', () => {
    expect(wrapper.vm.count).toBe(0);
  });

  it('increments the count when the increment button is clicked', () => {
    expect(wrapper.vm.count).toBe(0);
  });
});
```

Resources

- <https://vue-test-utils.vuejs.org/en/>
- GitHub repo with full source code from the demo

fin.