144-Testing the editor video

- create in imports/ui a file called Editor.test.js
- lets import the usual test modules

```
import { Meteor } from 'meteor/meteor';

import React from 'react';

import expect from 'expect';

import { mount } from 'enzyme';
```

- we are also going to import the component we are going to be testing

```
import { Meteor } from 'meteor/meteor';

import React from 'react';

import expect from 'expect';

import { mount } from 'enzyme';


import { Editor } from './Editor' ;
```

- **and also our notes fixtures to simulate**

```
import { Meteor } from 'meteor/meteor';

import React from 'react';

import expect from 'expect';

import { mount } from 'enzyme';


import { Editor } from './Editor' ;

import { notes } from '../fixtures/fixtures';
```

- **we then are going to check if we are in the client side**
- **then we want create a new describe block**

```
import { Meteor } from 'meteor/meteor';

import React from 'react';

import expect from 'expect';
```

```
import { mount } from 'enzyme';


import { Editor } from './Editor' ;
import { notes } from '../fixtures/fixtures';


if(Meteor.isClient){
    describe('Editor', function(){


    });
}
```

- **now is time to set up our test cases - we are going to be testing call and browserHistory, so lets first create two consts then later define them**

```
import { Meteor } from 'meteor/meteor';
import React from 'react';
import expect from 'expect';
import { mount } from 'enzyme';


import { Editor } from './Editor' ;
import { notes } from '../fixtures/fixtures';


if(Meteor.isClient){
    describe('Editor', function(){
        let broswerHistory;
        let call;


    });
}
```

- lets setup beforeEach and give and define the call variable

```
import { Meteor } from 'meteor/meteor';
import React from 'react';
```

```
import expect from 'expect';
import { mount } from 'enzyme';

import { Editor } from './Editor' ;
import { notes } from '../fixtures/fixtures';

if(Meteor.isClient){
    describe('Editor', function(){
        let broswerHistory;
        let call;

        beforeEach(function(){
            call = expect.createSpy();
        });
    });
}
```

- then we want browserHistory an object not a spy, the browserHistory should have a prop name push, and that should be spy

```
import { Meteor } from 'meteor/meteor';
import React from 'react';
import expect from 'expect';
import { mount } from 'enzyme';

import { Editor } from './Editor' ;
import { notes } from '../fixtures/fixtures';

if(Meteor.isClient){
    describe('Editor', function(){
        let broswerHistory;
        let call;
```

```
        beforeEach(function(){
            call = expect.createSpy();
            browserHistory = {
                push: expect.createSpy()
            }
        });
    });
}
```

- now we can move on the test cases

```
if(Meteor.isClient){
    describe('Editor', function(){
        let broswerHistory;
        let call;

        beforeEach(function(){
            call = expect.createSpy();
            browserHistory = {
                push: expect.createSpy()
            }
        });

        it('Should render pick note message', function(){

        });
    });
}
```

- lets now define wrapper

```
it('Should render pick note message', function(){
            const wrapper = mount(<Editor browserHistory={browserHistory} call={call}/>);
        });
```

- and we are going to expect that the wrapper has a paragraph with a text value to be some text

```
it('Should render pick note message', function(){
            const wrapper = mount(<Editor browserHistory={browserHistory} call={call}/>);


            expect(wrapper.find('p').text()).toBe('Pick or create a note to get started');
        });
```

- lets now startup our app in test mode, go to the terminal

```
joses-MacBook-Pro:notes mendoza$ npm test
```

- **the first should be passing**
- **and now we are going to be testing when there is no noteId and the message should say no note found**
- **so back in Editor.test.js**
- **so right we passed a prop that is not required selectedNoteId setting equal to our notes fixtures, and get the first note[0] and we get it's Id**

```
        it('Should render no note found message', function(){
            const wrapper = mount(<Editor browserHistory={browserHistory} call={call} selectedNo
teId={notes[0]._id}/>);
            expect(wrapper.find('p').text()).toBe('Note not found');
        });
```

- and that should pass our second test
- the next test case is going to verify that when we remove a note we call notes.remove and gets redirected to the dashboard

```
        it('Should render pick a note message', function(){
            const wrapper = mount(<Editor browserHistory={browserHistory} call={call}/>);
            expect(wrapper.find('p').text()).toBe('Pick or create a note to get started');
        });


        it('Should render no note found message', function(){
```

```
            const wrapper = mount(<Editor browserHistory={browserHistory} call={call} selectedNo
teId={notes[0]._id}/>);
            expect(wrapper.find('p').text()).toBe('Note not found');
        });

        it('Should remove note', function(){

        })
```

- **we the are going to define wrapper to an instance of Editor with the same props we used above but this time with another prop called note and we are going to set that prop equal to the notes fixture**

```
        it('Should remove note', function(){
            const wrapper = mount(<Editor browserHistory={browserHistory} call={call} selectedNo
teId={notes[0]._id} note={notes[0]}/>);

        })
```

- then we need to simulate a click on that button
- after that set up asserts for call spy and for push spy

```
        it('Should remove note', function(){
            const wrapper = mount(<Editor browserHistory={browserHistory} call={call} selectedNo
teId={notes[0]._id} note={notes[0]}/>);

            wrapper.find('button').simulate('click');
            expect(call).toHaveBeenCalledWith('notes.remove',notes[0]._id );
            expect(browserHistory.push).toHaveBeenCalledWith('/dashboard');

        })
```

-
-
-