125-Enzyme 101 video

- you can enzyme docs in https://github.com/airbnb/enzyme
- we are going to be using enzyme with mocha, and we are going to be adding enzyme with react addons
- so over to your terminal navigate to your notes directory

```
meteor npm install --save-dev enzyme@2.7.1 react-addons-test-utils@15.4.1
```

- now we can run tests on the client side, just like how we did the server testing , we are going to be creating files for every file we are going to be testing
- go to imports/ui and make a file called PrivateHeader.test.js
- now import the stuff we are going to render the private header component
- over at PrivateHeader.test.js

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

- now we are going to import the component that we are trying to test

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

import PrivateHeader from './PrivateHeader';
```

- **these** are goingt to be client side test, so first we need to check with an if statement to see if we are on the client side first

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

import PrivateHeader from './PrivateHeader';
```

```
if(Meteor.isClient) {

}
```

- **and now** can we add describe block, and we are going to be calling it the same as the component so it's easy to see over in the browser

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

import PrivateHeader from './PrivateHeader';

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

    })
}
```

- do our first test case, and this is just as an example, this would not be testing in a real app

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

import PrivateHeader from './PrivateHeader';

if(Meteor.isClient) {
    describe('PrivateHeader', function(){
        it('should set button text to logout', function(){

        });
    });
}
```

- then we are going to be calling mount() it takes one argument, it takes jsx, we are going to make an instance of <PrivateHeader/> and props it might require

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

import PrivateHeader from './PrivateHeader';

if(Meteor.isClient) {
    describe('PrivateHeader', function(){
        it('should set button text to logout', function(){
            mount(<PrivateHeader title="test title"/>);
        });
    });
}
```

- mount does return something so lets set equal to a variable

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

import PrivateHeader from './PrivateHeader';

if(Meteor.isClient) {
    describe('PrivateHeader', function(){
        it('should set button text to logout', function(){
           const wrapper = mount(<PrivateHeader title="test title"/>);

        });

    });

}
```

- wrapper is the building block for your enzymes insertions, lets think about enzyme like jquery, there are a lot similarities with the two, on this example we are using the find() method and we are targeting the button tag and we are also calling the text() value, so lets also give this a variable

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

import React from 'react';

import expect from 'expect';

import { mount } from 'enzyme';


import PrivateHeader from './PrivateHeader';


if(Meteor.isClient) {

    describe('PrivateHeader', function(){

        it('should set button text to logout', function(){

           const wrapper = mount(<PrivateHeader title="test title"/>);

           const buttonText = wrapper.find('button').text();

        });

    });

}
```

- now lets setup our assertion

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

import React from 'react';

import expect from 'expect';

import { mount } from 'enzyme';


import PrivateHeader from './PrivateHeader';


if(Meteor.isClient) {
    describe('PrivateHeader', function(){
        it('should set button text to logout', function(){
           const wrapper = mount(<PrivateHeader title="test title"/>);
           const buttonText = wrapper.find('button').text();

           expect(buttonText).toBe('Logout');
        });
    });
}
```

- over in the browser you should see the PrivateHeader section on the client side, so this passes
- to break it, just to see if it works, go to PrivateHeader.js and right something else Logout

```
import React from 'react';
import { Accounts } from 'meteor/accounts-base';

const PrivateHeader = (props)=>{
    return (
        <div className="header">
            <div className="header__content">
                <h1 className="header__title">{props.title}</h1>
                <button className="button button--link-text " onClick={()=>
Accounts.logout()}>dsLogout</button>
            </div>
        </div>
    );
}
```

- and it does fail test in the browser
- for our next test we are going to make sure our props.title get used

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


import PrivateHeader from './PrivateHeader';


if(Meteor.isClient) {
    describe('PrivateHeader', function(){
        it('should set button text to logout', function(){
           const wrapper = mount(<PrivateHeader title="test title"/>);
           const buttonText = wrapper.find('button').text();


           expect(buttonText).toBe('Logout');
        });


        it('should use title prop as h1 text', function(){
            const title = 'test title';
            const wrapper = mount(<PrivateHeader title={title}/>)
            const h1 = wrapper.find('h1').text();


            expect(h1).toBe(title);
        })
    });
}
```