137-Test Fixtures video

- lets start by copyin the const notes array from NoteList.test.js
- in the imports folder create another folder called fixtures and create a file called fixtures.js
- paste the notes array in there

```
const notes = [

    {

        _id: 'noteId1',

        title: 'Test title',

        body: '',

        updatedAt: 0,

        userId: 'userId1'

    }, {

        _id: 'noteId2',

        title: '',

        body: 'Something in here',

        updatedAt: 0,

        userId: 'userId2'

    }

];
```

- lets export this array

```
export const notes = [

    {

        _id: 'noteId1',

        title: 'Test title',

        body: '',

        updatedAt: 0,

        userId: 'userId1'

    }, {

        _id: 'noteId2',

        title: '',
        body: 'Something in here',

        updatedAt: 0,

        userId: 'userId2'

    }

];
```

- now delete the notes array from NoteList.test.js and lets import it instead

```
import React from 'react';

import expect from 'expect';

import  { mount } from 'enzyme';

import { Meteor } from 'meteor/meteor';


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

import { NoteList } from './NoteList';


if(Meteor.isClient){

    describe('NoteList', function(){

        it('shoud render NoteListItem for each note', function(){

            const wrapper = mount(<NoteList notes={notes}/>);


            expect(wrapper.find('NoteListItem').length).toBe(2);

            expect(wrapper.find('NoteListEmptyItem').length).toBe(0);

        });


        it('should render NoteListEmptyItem if zero notes', function(){

            const wrapper = mount(<NoteList notes={[]}/>);


            expect(wrapper.find('NoteListItem').length).toBe(0);

            expect(wrapper.find('NoteListEmptyItem').length).toBe(1);

        });

    });

}
```

- lets run the test suite and make sure that it still works
- now go to NoteListItem.test.js and import fixtures

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

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

if(Meteor.isClient){
```

- copy the value of const updateAt

```
 const updateAt = 1510584394273;
```

- **and paste onto fixtures**

```
export const notes = [
    {
        _id: 'noteId1',
        title: 'Test title',
        body: '',
        updatedAt: 1510584394273,
        userId: 'userId1'
    }, {
        _id: 'noteId2',
        title: '',
        body: 'Something in here',
        updatedAt: 1510584394273,
        userId: 'userId2'
    }
];
```

- now lets modify our code in NoteListItem.js
- first remove const title and const updatedAt from both test cases
- modify the NoteListItem component note prop

```
import React from 'react';

import expect from 'expect';

import  { mount } from 'enzyme';

import { Meteor } from 'meteor/meteor';


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

import { NoteListItem } from './NoteListItem';


if(Meteor.isClient){

    describe('NoteListItem', function (){

        it('should render title and time stamp', function (){

            const wrapper = mount(<NoteListItem note={notes[0]} />)


            expect(wrapper.find('h5').text()).toBe(notes[0].title);

            expect(wrapper.find('p').text()).toBe('11/13/17');

        });

        it('should set default title if no title set', function (){

             const wrapper = mount(<NoteListItem note={notes[1]} />)


            expect(wrapper.find('h5').text()).toBe('Untitled note');

        });

    });

}
```

- we are going to come up with a mocked Session object so we can spy on it

```
import React from 'react';

import expect from 'expect';

import  { mount } from 'enzyme';

import { Meteor } from 'meteor/meteor';
import { notes } from '../fixtures/fixtures';
import { NoteListItem } from './NoteListItem';

if(Meteor.isClient){

    let Session;

    beforeEach(()=>{

    })

    describe('NoteListItem', function (){
        it('should render title and time stamp', function (){
            const wrapper = mount(<NoteListItem note={notes[0]} />)

            expect(wrapper.find('h5').text()).toBe(notes[0].title);
            expect(wrapper.find('p').text()).toBe('11/13/17');
        });
        it('should set default title if no title set', function (){
             const wrapper = mount(<NoteListItem note={notes[1]} />)

            expect(wrapper.find('h5').text()).toBe('Untitled note');
        });
    });
}
```

- we are going to have to provide a Session.set property and that is going to be our spy
- so we are going to define our Session object, and our propery is going to be set and that is going to have the value of spy

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

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

if(Meteor.isClient){

    let Session;

    beforeEach(()=>{
        set: expect.createSpy()
    });

    describe('NoteListItem', function (){
        it('should render title and time stamp', function (){
            const wrapper = mount(<NoteListItem note={notes[0]} />)

            expect(wrapper.find('h5').text()).toBe(notes[0].title);
            expect(wrapper.find('p').text()).toBe('11/13/17');
        });
        it('should set default title if no title set', function (){
             const wrapper = mount(<NoteListItem note={notes[1]} />)

            expect(wrapper.find('h5').text()).toBe('Untitled note');
        });
    });
}
```

- now lets provide Session to the two components below

```
    describe('NoteListItem', function (){
        it('should render title and time stamp', function (){
            const wrapper = mount(<NoteListItem note={notes[0]} Session={Session} />)
            expect(wrapper.find('h5').text()).toBe(notes[0].title);
            expect(wrapper.find('p').text()).toBe('11/13/17');
        });
        it('should set default title if no title set', function (){
             const wrapper = mount(<NoteListItem note={notes[1]} Session={Session} />)

            expect(wrapper.find('h5').text()).toBe('Untitled note');
        });
    });
}
```

- now lets do anther test case below

```
   describe('NoteListItem', function (){
        it('should render title and time stamp', function (){
            const wrapper = mount(<NoteListItem note={notes[0]} Session={Session} />)

            expect(wrapper.find('h5').text()).toBe(notes[0].title);
            expect(wrapper.find('p').text()).toBe('11/13/17');
        });
        it('should set default title if no title set', function (){
             const wrapper = mount(<NoteListItem note={notes[1]} Session={Session} />)

            expect(wrapper.find('h5').text()).toBe('Untitled note');
        });

        it('should call set on click', function(){
            const wrapper = mount(<NoteListItem note={notes[0]} Session={Session} />);
            wrapper.find('div').simulate('click');

            expect(Session.set).toHaveBeenCalledWith('selectedNoteId',notes[0]._id);;
        })
```

```
    });
}
```

- lets no commit our changes in the terminal

```
joses-MacBook-Pro:notes mendoza$ git add .
```

then we commit and push to git hub

```
joses-MacBook-Pro:notes mendoza$ git commit -m "Integrate SelectedNoteId into list items and url"
joses-MacBook-Pro:notes mendoza$ git push
```