import React from 'react';

import PropTypes from 'prop-types';

/\_\*

\_ Render a list of items

\_

\_ @param {Object} props - List of items

\*/

function List(props) {

const { items } = props;

if (!items.length) {

// No Items on the list, render an empty message

return <span className="empty-message">No items in list</span>;

}

return (

<ul className="list-items">

{items.map(item => <li key={item} className="item">{item}</li>)}

</ul>

);

}

List.propTypes = {

items: PropTypes.array,

};

List.defaultProps = {

items: [],

};

export default List;

import React from 'react';

import { shallow } from '../enzyme';

import List from './List';

describe('List tests', () => {

it('renders list-items', () => {

const items = ['one', 'two', 'three'];

const wrapper = shallow(<List items={items} />);

// Expect the wrapper object to be defined

expect(wrapper.find('.list-items')).toBeDefined();

expect(wrapper.find('.item')).toHaveLength(items.length);

});

it('renders a list item', () => {

const items = ['Thor', 'Loki'];

const wrapper = shallow(<List items={items} />);

// Check if an element in the Component exists

expect(wrapper.contains(<li key='Thor' className="item">Thor</li >)).toBeTruthy();

});

it('renders correct text in item', () => {

const items = ['John', 'James', 'Luke'];

const wrapper = shallow(<List items={items} />);

//Expect the child of the first item to be an array

expect(wrapper.find('.item').get(0).props.children).toEqual('John');

});

});

import React from 'react';

import { shallow } from '../enzyme';

import List from './List';

describe('List tests', () => {

it('renders list-items', () => {

const items = ['one', 'two', 'three'];

const wrapper = shallow(<List items={items} />);

// Expect the wrapper object to be defined

expect(wrapper.find('.list-items')).toBeDefined();

expect(wrapper.find('.item')).toHaveLength(items.length);

});

it('renders a list item', () => {

const items = ['Thor', 'Loki'];

const wrapper = shallow(<List items={items} />);

// Check if an element in the Component exists

expect(wrapper.contains(<li key='Thor' className="item">Thor</li >)).toBeTruthy();

});

it('renders correct text in item', () => {

const items = ['John', 'James', 'Luke'];

const wrapper = shallow(<List items={items} />);

//Expect the child of the first item to be an array

expect(wrapper.find('.item').get(0).props.children).toEqual('John');

});

});