Snapshot-тестирование в React — это способ тестирования компонентов, который позволяет убедиться, что их представление (разметка или рендеринг) не изменилось непреднамеренно. Такой подход очень полезен при работе с интерфейсами, где точно нужно знать, что визуальное отображение компонента осталось консистентным.

**Как это работает?**

**Рендеринг компонента**: Компонент рендерится в определённом состоянии.

**Сохранение снимка**: Текущий рендер компонента сохраняется в виде снимка (snapshot) в файл. Эти файлы автоматически создаются (и обновляются) тестовыми библиотеками, такими как https://jestjs.io/.

**Сравнение с прошлым снимком**: При каждом запуске теста текущий рендеринг компонента сравнивается с сохранённым снимком.

**Если есть изменения, тест падает, указывая, что что-то поменялось.**

Как настроить snapshot-тест в React?

Убедитесь, что у вас установлен Jest. Если вы используете create-react-app, то Jest уже настроен по умолчанию.

Установите библиотеку для рендеринга компонентов в тестах, например, @testing-library/react.

**Пример:**

import React from 'react';

import { render } from '@testing-library/react';

import MyComponent from './MyComponent';

test('renders MyComponent correctly', () => {

  const { asFragment } = render(

    <MyComponent title="Hello, world!">

      <p>This is a child element</p>

    </MyComponent>

  );

  // Snapshot-тест проверяет, что рендер компонента не изменился

  expect(asFragment()).toMatchSnapshot();

});

**Как это работает?**

**Функция render**: "рендерит" компонент в виртуальный DOM.

**Метод asFragment**: возвращает виртуальный DOM в виде HTML-фрагмента.

**toMatchSnapshot:** сохраняет текущий результат в snapshot-файле (при первом прогоне теста) или сравнивает с уже сохранённым результатом.

При первом запуске будет создан файл **snapshots/MyComponent.test.js.snap** с примерно таким содержимым:

exports[`renders MyComponent correctly 1`] = `

<DocumentFragment>

  <div>

    <h1>Hello, world!</h1>

    <div>

      <p>This is a child element</p>

    </div>

  </div>

</DocumentFragment>

`;

**Когда тест может падать?**

Если структура компонента изменилась,

Если были модифицированы передаваемые пропсы,

Если стиль или структура внутреннего DOM-компонента изменились.

Когда тест "падает", нужно решить: изменение было ожидаемым или это баг.

Если изменение ожидалось, вы можете обновить снимок командой jest --updateSnapshot (или u при запуске тестов в интерактивном режиме).

Если изменение было нежелательным, надо внести исправления в код.