**1. Рендеринг компонента**

render\*\*: Рендерит React-компонент в виртуальное DOM-окружение, предоставляя функции для взаимодействия с этим DOM.

**2. Методы поиска элементов**

**🔍 Поиск по типам селекторов**

getByText\*\*: Ищет первый элемент, содержащий указанный текст. Синхронный метод, выбрасывает ошибку, если элемент не найден.

queryByText\*\*: Ищет первый элемент с указанным текстом. Синхронный метод, возвращает null, если элемент не найден.

findByText\*\*: Асинхронно ищет первый элемент с указанным текстом. Ожидает, пока элемент появится в DOM.

getByRole\*\*: Ищет элемент по его роли (например, кнопка, чекбокс, заголовок). Синхронный метод, выбрасывает ошибку, если элемент не найден.

queryByRole\*\*: Ищет элемент по роли. Возвращает null, если элемент не найден.

findByRole\*\*: Асинхронно ищет элемент по роли. Используется для поиска элемента, который появляется в DOM со временем.

getByPlaceholderText\*\*: Ищет элемент по атрибуту placeholder. Синхронный метод.

queryByPlaceholderText\*\*: Ищет элемент по атрибуту placeholder. Возвращает null, если элемент не найден.

findByPlaceholderText\*\*: Асинхронно ищет элемент по атрибуту placeholder.

getByLabelText\*\*: Ищет элемент, связанный с label. Обычно используется для полей ввода. Синхронный метод.

queryByLabelText\*\*: Ищет элемент, связанный с label. Возвращает null, если элемент не найден.

findByLabelText\*\*: Асинхронно ищет элемент, связанный с label.

getByTestId\*\*: Ищет элемент по пользовательскому атрибуту data-testid. Синхронный метод.

queryByTestId\*\*: Ищет элемент по атрибуту data-testid. Возвращает null, если элемент не найден.

findByTestId\*\*: Асинхронный метод, ищет элемент по атрибуту data-testid.

**🔎 Поиск нескольких элементов**

getAllBy\*: Возвращает массив всех элементов, соответствующих селектору. Выбрасывает ошибку, если элементов нет.

queryAllBy\*: Возвращает массив всех элементов, соответствующих селектору, или пустой массив, если элементы не найдены.

findAllBy\*: Асинхронно возвращает массив всех элементов, соответствующих селектору.

**3. Работа с screen**

screen\*\*: Предоставляет глобальный доступ ко всем методам поиска (например, screen.getByText), чтобы не обращаться к возвращаемому объекту из render.

**4. Утилиты для тестов**

fireEvent\*\*: Симулирует пользовательские события, включая клики, ввод текста, наведение мыши и другие интерфейсные взаимодействия.

userEvent\*\*: Более "реалистичная" и расширенная библиотека для симуляции событий пользователя (например, сложных событий, таких как ввод текста или копирование/вставка).

waitFor\*\*: Асинхронная утилита, ожидает, пока определенные изменения или элементы появятся в DOM.

act\*\*: Помогает тестировать React-компоненты, оборачивая код для выполнения рендеров и обновлений состояния.

cleanup\*\*: Удаляет все рендеры из DOM между тестами. Обычно вызывается автоматически.

**5. Матчеры для проверки**

⚡ Основные матчеры из jest-dom:

toBeInTheDocument\*\*: Подтверждает, что элемент существует в DOM.

toHaveTextContent\*\*: Проверяет, что элемент содержит указанный текст.

toHaveAttribute\*\*: Проверяет, что у элемента есть определенный атрибут и, опционально, его значение.

toHaveValue\*\*: Проверяет значение value у элемента, например, у полей формы.

toBeDisabled и toBeEnabled\*\*: Проверяют, что элемент заблокирован или доступен для взаимодействия.

toBeEmptyDOMElement\*\*: Проверяет, что элемент пустой (не содержит дочерние элементы).

toBeVisible\*\*: Проверяет, что элемент видим пользователю (например, он не скрыт через CSS).

toHaveClass\*\*: Проверяет наличие указанного имени класса у элемента.

toContainElement\*\*: Проверяет, что один элемент содержит другой элемент.

toHaveFocus\*\*: Проверяет, что элемент находится в фокусе.

**⚡ Общие матчеры из Jest:**

toEqual\*\*: Проверяет равенство объектов или значений.

toBe\*\*: Проверяет строгое равенство.

toMatchSnapshot\*\*: Делает "снимок" состояния компонента и сравнивает его с предыдущими тестами.

toThrow\*\*: Проверяет, что вызов функции выбрасывает ошибку.

**6. Асинхронные действия**

async/await\*\*: Используется для тестирования асинхронных изменений в DOM, например, при ожидании данных от API.

waitForElementToBeRemoved\*\*: Асинхронная утилита, проверяющая исчезновение элемента из DOM.