



# Native Testing Library

[github.com/testing-library/native-testing-library](https://github.com/testing-library/native-testing-library)

Simple and complete cheat sheet

## render a component

```
import { render } from 'native-testing-library';

// See render result for more details...
const result = render(<TextInput />);
```

## search for elements

```
const { getByText } = render(<Text>hello</Text>);

// Check out jest-native on npm for handy assertions...
const element = getByText('hello');
```

## fire an event

```
import { fireEvent } from 'native-testing-library';

fireEvent.press(element);
// Or you can fire events manually with...
fireEvent(element, new NativeTestEvent('press'));
```

## see debug output

```
const { debug } = render(<Text>hello</Text>);

// Pretty print native elements
debug();
```

## wait for any assertion

```
import { wait } from 'native-testing-library';

// Retry search every 50ms for 4500ms
wait(() => getByText('async result'));
```

## search variants

(return value)

|                   |                             |
|-------------------|-----------------------------|
| <b>getBy</b>      | Element or Error            |
| <b>getAllBy</b>   | Element[] or Error          |
| <b>queryBy</b>    | Element or null             |
| <b>queryAllBy</b> | Element[] or []             |
| <b>findBy</b>     | Promise<Element> or Error   |
| <b>findAllBy</b>  | Promise<Element[]> or Error |

## search types

|                        |                                   |
|------------------------|-----------------------------------|
| <b>LabelText</b>       | accessibilityLabel="go back"      |
| <b>PlaceholderText</b> | placeholder="username"            |
| <b>Text</b>            | <Text>About</Text>                |
| <b>HintText</b>        | accessibilityHint="movie poster"  |
| <b>Title</b>           | <Button title="hello" />          |
| <b>DisplayValue</b>    | Current value of Picker or Switch |
| <b>Role</b>            | accessibilityRole="button"        |
| <b>TestId</b>          | testID="username-input"           |

## text matches

```
const { getByText } = render(<Text>Hello World</Text>);

getByText('Goodbye World'); // ✗
getByText(/hello world/); // ✗
getByText('ello Worl', { exact: false }); // ✓
getByText('Hello World'); // ✓
```

## wait for appearance

```
test('movie title appears', async () => {
  // element is initially not present...
  expect(() => getByText('aladdin')).toThrow();

  // wait for appearance
  await wait(() => {
    expect(getByText('aladdin')).toBeTruthy();
  });

  // wait for appearance and return the result
  await waitForElement(() => getByText('aladdin'));
});
```

## assert for absence

```
expect(queryByText('submit')).toBeNull();
```

## the render result

(description)

|                       |                               |
|-----------------------|-------------------------------|
| <b>container</b>      | View containing your UI       |
| <b>baseElement</b>    | Root View of the render       |
| <b>debug(element)</b> | Pretty print native elements  |
| <b>unmount()</b>      | Unmount your component        |
| <b>rerender(ui)</b>   | Render again at the container |
| <b>...queries</b>     | Queries for the baseElement   |

## scope your queries

```
import { within } from 'native-testing-library';

within(loginForm).getByText('submit');
```