Version: 3.x

Testing with Jest

> ⓘ INFO
>
> This page was ported from an old version of the documentation.
>
> As we're rewriting the documentation some of the pages might be a little outdated.

Reanimated test mocks use web implementation of Reanimated2. Before you begin using Reanimated mocks you need some setup actions.

Setup

First, make sure that your tests run with Node version 16 or newer.

Add the following line to your `jest-setup.js` file:

```
require('react-native-reanimated').setUpTests();
```

`setUpTests()` can take optional config argument. Default config is `{ fps: 60 }`, setting framerate to 60fps.

To be sure, check if your `jest.config.js` file contains:

```
...
preset: 'react-native',
setupFilesAfterEnv: ['./jest-setup.js'],
...
```

> ⓘ CAUTION

> If you use Jest in a version **older than 28**, you should set `setupFiles` property instead of `setupFilesAfterEnv`

If you have custom babel configuration for testing, make sure that Reanimated's babel plugin is enabled in that environment.

API

Style checker

- Checking equality of selected styles with current component styles

  `expect(component).toHaveAnimatedStyle(expectedStyle)`

  `component` - tested component `expectedStyle` - contains expected styles of testing component, for example `{ width: 100 }`

- Checking equality of all current component styles with expected styles

  `expect(component).toHaveAnimatedStyle(expectedStyle, {exact: true})`

- You can get all styles of tested component by using `getDefaultStyle`

  `getDefaultStyle(component)`

  `component` - tested component

Timers

You can use Jest's fake timers to control animation progress. Check the full guide about mocking timers on Jest documentation website.

```
jest.useFakeTimers();
// call animation
jest.runAllTimers();
```

If you want more control over animation, you can use `jest.advanceTimersByTime` to move to a certain point in the animation:

```
jest.useFakeTimers();
// call animation
jest.advanceTimersByTime(250);
// make assertions on what you expect the styles of a component should be after
250ms
```

Example

The below code shows an example of test that runs a 250ms of animation and verifies the component style after that point in time.

```
// Setup fake timers - this can be done before the tests are run
jest.useFakeTimers();

test('stop in the middle of animation', () => {
  const style = { width: 0 };

  const { getByTestId } = render(<AnimatedComponent />);
  const view = getByTestId('view');
  const button = getByTestId('button');

  expect(view.props.style.width).toBe(0);
  expect(view).toHaveAnimatedStyle(style);

  fireEvent.press(button);
  jest.advanceTimersByTime(250); // if whole animation duration is a 500ms
  style.width = 50; // value of component width after 250ms of animation
  expect(view).toHaveAnimatedStyle(style);
});
```

Check links below for full examples of tests from Reanimated repo

- SharedValue.test.tsx
- Animation.test.tsx

Recommended testing library

- @testing-library/react-native

- @testing-library/react-hooks - for dealing with hooks

Edit this page