





Vitalist


Desenvolvido por Vite, o Vitest também possui um servidor de desenvolvimento sob o capô ao executar os testes. Isso permite que o Vitest forneça uma bela interface do usuário para visualizar e interagir com seus testes


Me checking my test results



 Vitest



 Search...



PASS (2)

✓ src/App.tsx

3ms

✓ src/App.test.tsx

127ms

✓ App

126ms

✓ Button be defined

36ms

✓ Fetch user if clicked on button

88ms

✓ /Users/ziegler-dev/projects/react-conf-lib/apps/my-app/src/App.test.tsx

Report

Module Graph

Code

Console (0)

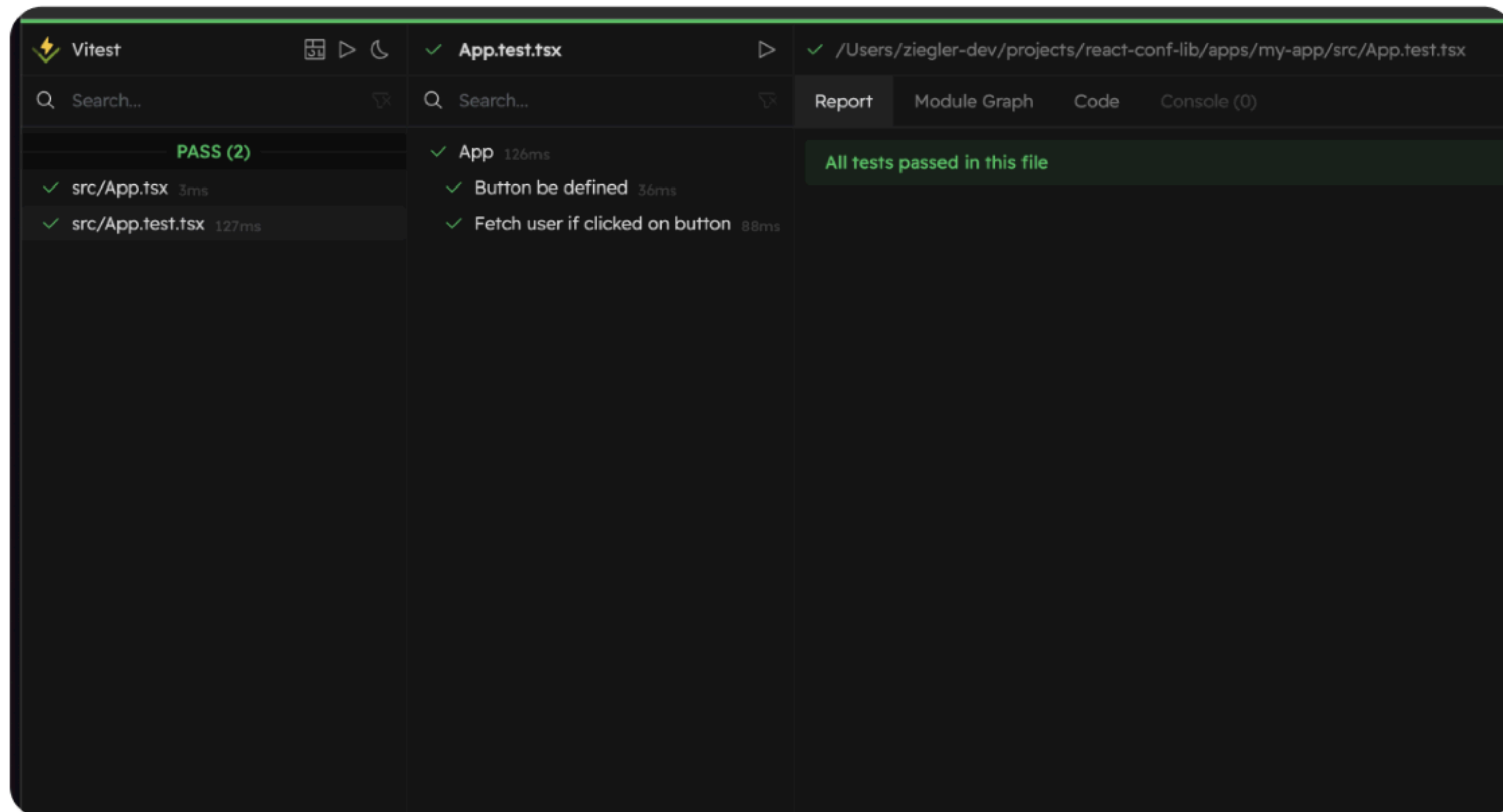
All tests passed in this file

Me checking my test results



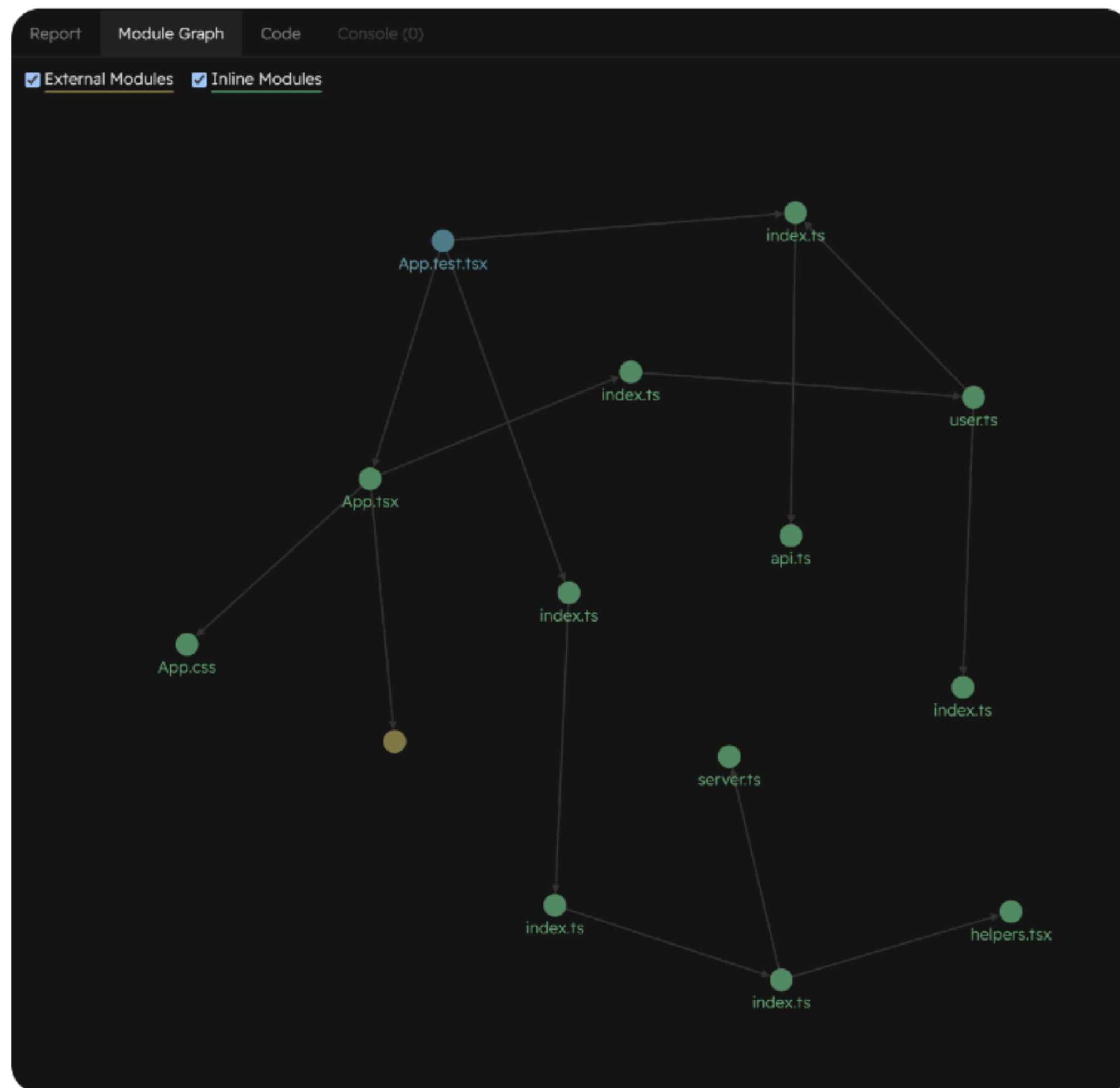
Vitest UI

Desenvolvido por Vite, o Vitest também possui um servidor de desenvolvimento sob o capô ao executar os testes. Isso permite que o Vitest forneça uma bela interface do usuário para visualizar e interagir com seus testes



Vitest UI

- Module Graph - Report Status - Runner Code Editor



Search... Report Module Graph Code Console (0)

```
✓ App 124ms
✓ Button be de
✓ Fetch user if
```

```
1 import { describe, expect, test, beforeEach } from 'vitest';
2 import {
3   render,
4   screen,
5   userEvent,
6   createMockServer,
7   createRestHandler,
8   startMockServer,
9 } from 'shared';
10 import App from './App';
11 import { apiRoutes } from './routes';
12
13 interface LocalTestContext {
14   buttonLabel: string;
15 }
16
17 describe('App', () => {
18   setupTests();
19   beforeEach<LocalTestContext>(async (context) => {
20     // typeof context is 'TestContext & LocalTestContext'
21     context.buttonLabel = 'Fetch User';
22   });
23   test<LocalTestContext>('Button be defined', async (context) => {
24     render(<App />);
25
26     const button = await screen.findByText(context.buttonLabel);
27     expect(button).toBeDefined();
28   });
29   test<LocalTestContext>('Fetch user if clicked on button', async (context) => {
30     const wrapper = render(<App />);
31
32     const button = wrapper.getByText(context.buttonLabel);
33     userEvent.click(button);
34     await wrapper.findByText(/john/i);
35   });
36 });
```