



Testing Asyncrous Code is Hard

● 模倣品同歩測定式

● 懷境複齋不好模稜，且容易出錯

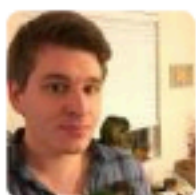
● 出錯時不知道要按誰

● 精 確 測 試 同 步

Testing Asynchronous Code is Hard

- 模擬非同步測試
 - 情境複雜不好模擬，且容易出錯
 - 出錯時不知道要找誰
- 精確非同步測試





rimunroe commented on 3 May • edited



I'm still seeing the same issue with it in v4.17.4 of Lodash. I tried it with Jest 18 & 19 on Node 4 & 6.

```
jest.useFakeTimers();

var _ = require('lodash');

test('fails to mock Lodash timers correctly', function() {
  var test = jest.fn();
  var debounced = _.debounce(test, 1000);

  debounced();
  debounced();

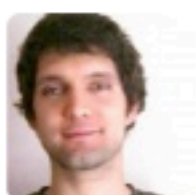
  jest.runAllTimers();

  expect(test).toHaveBeenCalledTimes(1);
});
```

Running that with Jest causes it to say "Ran 100000 timers, and there are still more! Assuming we've hit an infinite recursion and bailing out...". Do you think this is a Lodash issue, or is this something I should pursue on Jest's end?



1



jdalton commented on 4 May

Owner



Lodash doesn't hold on to references to `setTimeout`, `clearTimeout`, or `Date.now` so your guess is as good as mine.