

# Sinon stub ES6模块

## 场景一：stub外部模块依赖

模块测试中经常遇到的场景是引入外部模块依赖，在写单元测试时我们应该尽量减少模块依赖，只关心模块自身的执行路径，对于子模块的细节应该在子模块的测试中进行覆盖。

*foo.ts*

```
1. export foo(){
2.   // do something like fetch XHR等
3. }
```

*bar.ts*

```
1. import {foo} from './foo'
2. export bar(state){
3.   if(state){
4.     foo(state)
5.   }else{
6.     // do something
7.   }
8. }
```

我们需要测试`bar.ts`的 `bar` 模块,但是在 `bar` 模块内部调用了 `foo` 模块，在这里我们只关心 `foo` 模块是否以正确的参数被调用，而不关心 `foo` 模块真正做了什么，因此我们需要 `stub foo` 这个外部模块依赖。

*bar.test.ts*

```

1. import {expect} from 'chai';
2. import * as sinon from 'sinon';
3.
4. import * as fooModule from '../foo/foo'; // 必须以import
   * as 导出到对象上
5. import {bar} from './bar'
6.
7. describe('测试bar',()=>{
8.     it('bar正确调用',()=>{
9.         const stub=sinon.stub(fooModule,'foo') // stub f
   oo模块
10.         bar('aaa');
11.         expect(stub.calledWith('aaa')).to.be.true
12.     })
13. })

```

## 场景二：stub模块内部依赖

场景一中模块依赖来自于外部，但是模块内部依赖是更加常见的场景，例如下面这样，一个模块引用了另一个内部子模块。

*bar1.ts*

```

1.
2. function foo{
3.
4. }
5. export bar(state){
6.     if(state){
7.         foo(state)
8.     }else{
9.         // do someting
10.    }
11. }

```

*bar2.ts*

```

1.
2. export function foo{
3.
4. }
5. export bar(state){
6.     if(state){
7.         foo(state)
8.     }else{
9.         // do someting
10.    }
11. }

```

- 在`bar1`中 `foo` 作为完全封闭的内部函数，只能在 `bar` 中进行覆盖测试，暂时没有办法stub。
- 在`bar2`中 `foo` 作为外部模块导出，可以进行单独的测试覆盖，在 `bar` 模块中就可以将 `foo` 看作一个黑盒，不必关心 `foo` 内部实现，其内部实现已经在针对 `foo` 的测试用例中进行了覆盖，因此考虑stub内部依赖 `foo`。

尝试使用场景一中的方法进行stub，但是并不能实现，因为ES6模块被编译为commonJS模块时

```

1. export function foo(){}

```

被编译成：

```

1. "use strict";
2. exports.__esModule = true;
3. function foo() { }
4. exports.foo = foo;

```

我们使用`sinon.stub(fooModule,'foo')`时只是stub了 `exports.foo`，但是在模块内部 `bar` 调用的时候调用的是 `foo`，两者其实不是同一个方法了。

查了很多资料，都没有办法优雅的解决这个问题，比较多的解决方式是使用 `babel-plugin-rewire-exports` 和 `babel-plugin-rewire` 两个插件，通过类似于angular的注入，可以在外部调用注入的方法更改模块内部，达到stub的目的，但是这种方式破坏性太大。

这种方式的使用可以参考[在 es6 中 sinon 的正确打开方式](#)

目前stub ES6内部模块暂时没有比较好的办法。

## 参考

- [Improve ability to stub/mock exports of ES2015 modules transpiled by Babel \(T6748\)](#)
- [在 es6 中 sinon 的正确打开方式](#)
- [Mocking ES6 module import with and without Dependency Injection](#)
- [Stub private functions in an ES6 module](#)