Sinon JS

If your test will also involve other 3rd party library, etc. multer. You better also check their own test cases for reference. e.g. https://github.com/expressjs/multer/tree/master/test

Reference

- https://sinonjs.org/
- https://semaphoreci.com/community/tutorials/best-practices-for-spies-stubs-and-mocks-in-sinon-js
- https://codeutopia.net/blog/2016/05/23/sinon-js-quick-tip-how-to-stubmock-complex-objects-such-as-dom-objects/

Spy

- Check if a method is called

```
const spy = sinon.spy(object, 'method')
...
spy.restore() // remove spy
expect(spy.callCount).to.be.eql(n) // check if the method is call n times
```

Stub

- Replace existing method with stub to avoid actually execute the method

Method 1

- define a fake function so that it will call that fake one when call

```
function fake() {  // a fake function to be called
}

const stub = sinon.stub(object, 'method').callFakes(fake) // you could also
replace the fake with another real function
...
stub.restore() // remove stub
expect(stub.callCount).to.be.eql(n) // check if the method is call n times
```

Method 2

- define a return value no matter how you call the stub

```
const stub = sinon.stub(object, 'method').returns(val) // always return same
val
...
const test = stub()
expect(test).to.be.eql(val)
stub.restore() // remove stub
expect(stub.callCount).to.be.eql(n) // check if the method is call n times
```

Method 3

- resolve a promise with same value no matter how you call the stub

```
const stub = sinon.stub(object, 'method').resolve(val) // always resolve
same val
...
const test = await stub()
expect(test).to.be.eql(val)
stub.restore() // remove stub
expect(stub.callCount).to.be.eql(n) // check if the method is call n times
```

From:
https://wiki.questwork.com/ - **Questwork's Wiki**

Permanent link:
https://wiki.questwork.com/doku.php?id=development:testing:sinonjs:start

Last update: **2019/01/13 05:05**