

七牛前端测试实践

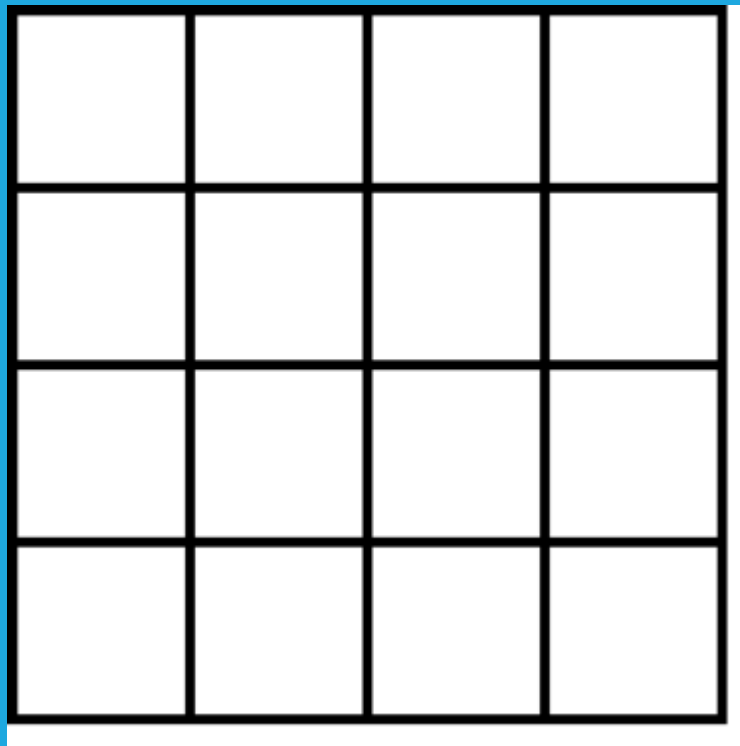


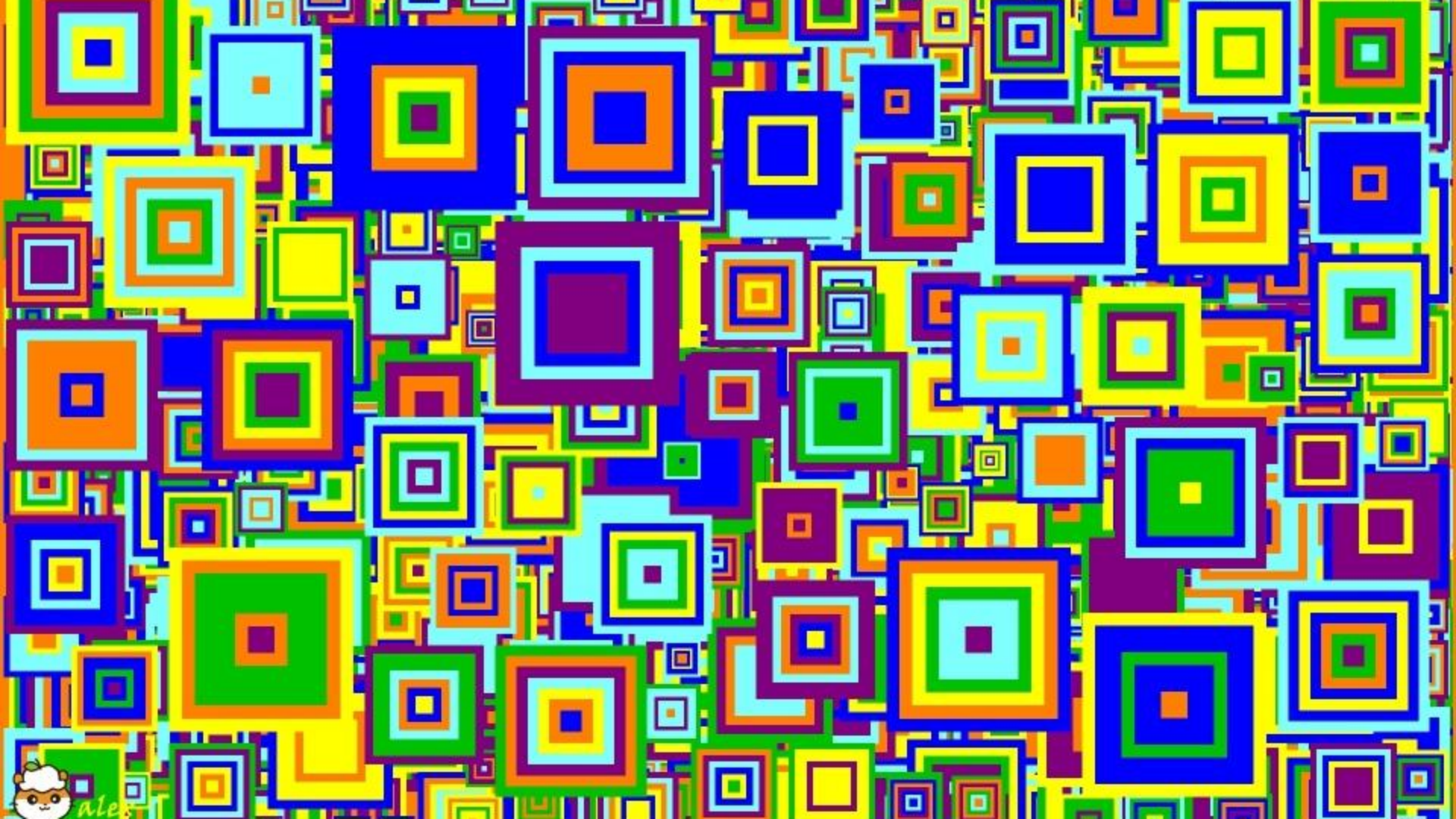
马逸清



我们的代码

理想中





alex

代码越来越乱

一开始是没有测试的

动一动代码产生 bug

代码放久了，不敢重构



早期尝试

selenium + phantom

模拟用户点点点，自动化端到端测试

```
describe '用户正常登录', ->
  before (done) ->
    driver.findElement(wd.By.name('username')).sendKeys(testAccount.username)
    .then -> driver.findElement(wd.By.name('password')).sendKeys(testAccount.password)
    .then -> driver.findElement(wd.By.css('#loginBtn')).click()
    .then -> done()
  after (done) ->
    driver.findElement(wd.By.css('.account-info')).click()
    .then -> driver.findElement(wd.By.css('.logout')).click()
    .then -> done()

it '应该跳转到首页', (done) ->
  driver.getTitle()
  .then (title) ->
    assert.equal title, '首页 - 七牛云存储'
    done()

it '应该正确显示账户信息', (done) ->
  driver.findElement(wd.By.css('.account-info')).getText()
  .then (user) ->
    assert.equal user, testAccount.username
    done()
```

工作量比较大, 难以坚持

早期尝试

测试 基础函数

但是对于 DOM 有依赖的部分不好测试



选择一个空间 ▾



首页



MPS



服务市场



统计



财务



账号

账户信息

用户类型: 标准用户

账户余额: ¥76.65

充值

使用概况

2015年06月01日 - 2015年06月30日

文件存储

869.24 MB

下载流量

217.80 MB

API请求

1293 次



云助开源

七牛开源项目扶持计划



androidday201

4.768MB

0B

6月8日 6月22日

文件存储

134.08 KB

下载流量

0 B

API请求

0 次



devfest

4.768MB

0B

6月8日 6月22日

文件存储

16.42 MB

下载流量

0 B

API请求

0 次

```
describe('$(el).setActive()', function() {
  describe('set top nav bar active status', function() {
    var INDEX_BOX_MENU, BUCKET_BOX_MENU;
    var resetTestDom = function() {
      INDEX_BOX_MENU = $('<div class="right-op index-box-op">\
        <ul id="box-right-ops" class="ops">\
          <a href="/" class="op" ><li class=""><i class="icon-small-home"></i>首页</li></a>\
          <a href="/mps/pipeline/intro" class="op" data-catalog="/mps"><li class=""><i class="icon->\
          <a href="/report" class="op" ><li class=""><i class="icon-form"></i>报表中心</li></a>\
          <a href="/financial" class="op" ><li class=""><i class="icon-finance"></i>财务</li></a>\
          <a href="/setting" class="op" ><li class=""><i class="icon-setting"></i>帐号设置</li></a>\
        </ul>\
      </div>');
      BUCKET_BOX_MENU = $('<div class="right-op bucket-box-op">\
        <ul id="box-right-ops" class="ops">\
          <a href="/bucket/test/index" class="op" ><li class=""><i class="icon-box"></i>空间</li></a>\
          <a href="/bucket/test/resource" class="op" ><li class=""><i class="icon-cloud"></i>内容管理</li></a>\
          <a href="/bucket/test/processor" class="op" ><li class=""><i class="icon-pencils"></i>数据</li></a>\
          <a href="/bucket/setting/?bucket=test" class="op" ><li class=""><i class="icon-setting"></i></li></a>\
        </ul>\
      </div>');
    };

    beforeEach(function() {
      resetTestDom();
    });

    describe('should set top nav active correct with data-catalog attr as prefix', function() {
      it('/mps/pipeline', function() {
        var path = '/mps/pipeline';
        INDEX_BOX_MENU.find('.op', '#box-right-ops').setActive('current', path);
        INDEX_BOX_MENU.find('.current').should.have.length(1);
        INDEX_BOX_MENU.find('.current').attr('href').should.eql('/mps/pipeline/intro');
```



ANGULARJS

隔离 DOM

注意力集中到 逻辑 部分

注入的方式

使依赖的处理更加方便

utils

model 状态

view 状态

Utils

```
validator = undefined

beforeEach module 'app.utils'

beforeEach inject [
  'validatorUtil'
  (validatorUtil) -> validator = validatorUtil
]

describe 'mirrorSource', ->

  it 'should valid when host valid', ->
    validator.mirrorSource('http://portal.demo.qiniu.io').should.be.true
    validator.mirrorSource('http://portal.demo.qiniu.website').should.be.true

  it 'should invalid when host is invalid', ->
    validator.mirrorSource('http://qiniu_test.com').should.be.false

  it 'should valid when url with protocol http or https', ->
    validator.mirrorSource('http://www.qiniu.com').should.be.true
    validator.mirrorSource('https://www.qiniu.com').should.be.true
```

```
19 mockDownloadTokenResp = {
20   "code": 200
21   "data": {
22     "download_token": "MHtq7sCV4ih5wLdMiEVHiwFaynXRzqJ63-DF6-dv:NTQ3YWI5N2E5
23   }
24 }
25
26 beforeEach module 'app.service'
27
28 beforeEach inject [
29   '$httpBackend'
30   'policyService'
31
32   ($httpBackend, policyService) ->
33     httpBackend = $httpBackend
34     policy = policyService
35 ]
36
37 beforeEach ->
38   httpBackend.whenGET '/api/policy/upload-token'
39     .respond 200, mockUploadTokenResp
40
41   httpBackend.whenGET '/api/policy/download-token'
42     .respond 200, mockDownloadTokenResp
43
44 afterEach ->
45   httpBackend.verifyNoOutstandingExpectation()
46   httpBackend.verifyNoOutstandingRequest()
47
48 describe '@uploadToken', ->
```

service

```
41  
42 beforeEach -> module 'app.service', 'app.controller', 'ui.router'  
43  
44 beforeEach inject [  
45   '$controller'  
46   '$rootScope'  
47   '$httpBackend'  
48  
49   ($controller, $rootScope, $httpBackend) ->  
50     httpBackend = $httpBackend  
51  
52     bucketReqHandler = httpBackend.when 'GET', '/api/bucket'  
53     .respond 200, mockBucketsResp  
54  
55     scope = $rootScope.$new()  
56  
57     controller = $controller  
58   ]  
59
```

controller

```
69 describe 'scope.sortByName', ->
```

```
70  
71   beforeEach ->
```

```
72     controller 'homeController', '$scope': scope  
73     httpBackend.flush()  
74
```

```
75   it 'should sorted by name asc default', ->
```

```
76  
77     scope.sortByName()
```

```
78  
79     buckets = scope.buckets
```

```
80     for i in [1...buckets.length]
```

```
81       buckets[i - 1].name.should.not.above buckets[i].name  
82
```

```
83   it 'should sorted by name desc when set desc to true', ->
```

```
84  
85     scope.sortByName(true)
```

```
86  
87     buckets = scope.buckets
```

```
88     for i in [1...buckets.length]
```

```
89       buckets[i - 1].name.should.not.below buckets[i].name  
90
```

controller

依赖不好处理情况

容易想到的是 mock module

另一种方式是用 stub 来处理依赖

```
describe '@isExpire', ->
```

```
  NOW = (new Date('2015-01-01T15:30:30+08:00')).getTime()
```

```
  before ->
```

```
    sinon.stub($, 'now').returns NOW
```

```
  it 'should expired 10 second before expire time', ->
```

```
    expires = [
```

```
      '2015-01-01T15:30:20+08:00'
```

```
      '2015-01-01T15:30:30+08:00'
```

```
      '2015-01-01T15:30:39+08:00'
```

```
      '1970-01-01T00:00:00+08:00'
```

```
    ]
```

```
    for expire in expires
```

```
      session.set ACCESS_TOKEN, REFRESH_TOKEN, expire
```

```
      session.isExpire().should.be.true
```

```
  it 'should not expire before expire time', ->
```

```
    expires = [
```

```
      '2015-01-01T15:30:40+08:00'
```

```
      '2015-01-01T16:30:40+08:00'
```

```
    ]
```

```
    for expire in expires
```

```
      session.set ACCESS_TOKEN, REFRESH_TOKEN, expire
```

```
      session.isExpire().should.be.false
```

stub 和 mock 的区别

stub 适合处理仅关注返回值的依赖

mock 关注依赖的行为验证, 如无返回的依赖

关注行为验证的 stub 叫 spy

mock data

```
11 mockBucketsResp = {
12     "code": 200,
13     "data": [{
14         "name": "qiniu-image",
15         "files": 9032,
16         "space": 10.5795,
17         "bandwidth": 39.4322,
18         "last_operation": "申请开启防盗链黑名单",
19         "last_operated_at": "2013-12-25T15:23:23"
20     }, {
21         "name": "ma-test",
22         "files": 9032,
23         "space": 10.5795,
24         "bandwidth": 39.4322,
25         "last_operation": "申请开启防盗链黑名单",
26         "last_operated_at": "2013-12-27T23:23:23"
27     }, {
28         "name": "unprotect"
```

```
beforeEach inject [  
  '$controller'  
  '$rootScope'  
  '$httpBackend'
```

```
($controller, $rootScope, $httpBackend) ->  
  httpBackend = $httpBackend
```

```
    bucketReqHandler = httpBackend.when 'GET', '/api/bucket'  
      .respond 200, mockBucketsResp
```

```
    scope = $rootScope.$new()  
    scope.$state = {}  
    scope.$stateParams = {}
```

```
    scope.$state.go = (state, params) ->  
    scope.$stateParams.bucketName = 'ma-test'
```

```
    controller = $controller
```

```
]
```

fixture

load html and json

```
it 'should parse bucket states', ->  
  b = fixture.load "/src/qiniu.com/portal/front/mock/buckets/bucket.1.json"  
  bucket = bucketService.newBucket b.data
```

目前用到的



+



+



or



karma-coverage + karma-fixture + karma-coffee

```
config.set
```

```
frameworks: ['mocha', 'sinon-chai', 'fixture']
```

```
phantomjsLauncher:
```

```
  cmd:
```

```
    linux: 'phantomjs'
```

```
    darwin: 'phantomjs'
```

karma config

```
preprocessors:
```

```
'src/qiniu.com/portal/front/scripts/**/*.!(*Test).coffee': ['coverage']
```

```
'src/qiniu.com/portal/front/scripts/_def.coffee': 'coffee'
```

```
'src/qiniu.com/portal/front/scripts/**/*.!(*Test).coffee': 'coffee'
```

```
'src/qiniu.com/portal/front/scripts/**/*.Test.coffee': 'coffee'
```

```
'src/qiniu.com/portal/front/mock/**/*.json': 'json_fixtures'
```



```
files: [  
  "#{BOWER_PATH}/jquery/dist/jquery.js"  
  
  "#{BOWER_PATH}/angular/angular.js"  
  "#{BOWER_PATH}/angular-ui-router/release/angular-ui-router.js"  
  "#{BOWER_PATH}/angular-local-storage/dist/angular-local-storage.js"  
  "#{BOWER_PATH}/angular-mocks/angular-mocks.js"  
  "#{BOWER_PATH}/deep-diff/releases/deep-diff-0.3.2.min.js"  
  
  'src/qiniu.com/portal/front/scripts/_def.coffee'  
  'src/qiniu.com/portal/front/scripts/**/*.!(Test).coffee'  
  'src/qiniu.com/portal/front/scripts/**/*.Test.coffee'  
  
  'src/qiniu.com/portal/front/mock/**/*.json'  
]  
  
reporters: [  
  'spec'  
  'coverage'  
]  
  
coverageReporter:  
  type: 'html'  
  dir: 'coverage/front/'
```

karma config

配置 gulp

```
1  karma = require 'karma'
2
3  module.exports = (gulp, opt, $) ->
4
5    testOnce: (done) ->
6      karma.server.start
7        configFile: __dirname + '/../karma.conf.coffee'
8      , done
9
10   testAndWatch: (done) ->
11     karma.server.start
12       configFile: __dirname + '/../karma.conf.coffee'
13       singleRun: false
14       autoWatch: true
15     , done
16
```

```
testTasks = require('./gulp-tasks/test.coffee')(gulp, opt, plugins)
```

```
gulp.task 'test', testTasks.testOnce
```

```
gulp.task 'testing', testTasks.testAndWatch
```

后续

持续集成
覆盖率报告

mock data 的改进
UI 自动化测试

实际体会到的好处

找出没有察觉的 bug

加深对问题的理解

降低修改代码的心理负担

前端 MVC 框架展示和逻辑分离

Node 的发展产生了完善的工具

搭建测试环境更加容易了



YEOMAN

这样就可以愉快的测试了么

迭代速度快

业务需求不稳定

人就那么些

理想和现实之间的平衡

实际情况比解决方案复杂

测试重要的部分 - 评估出错的风险

测试稳定的部分 - 分离模块

测试依赖少的部分 - 良好的设计

养成验证的习惯

Q&A

<https://plus.google.com/+马逸清>

