单元测试与自动化

李智维

简介

- 李智维
- 21 岁
- 中学参加算法竞赛
- 大二休学在 LeanCloud 工作了一年半，联合创立了 Reviewcode.cn，和聚美副总裁创业过一段时间
- Android、iOS、前端、后端都做过一些线上产品
- 现在创业做「趣直播」，知识直播平台，上线一个月，用户5200，流水 4 万，主播 9 成分成

一个测试例子

保存一个对象到 LeanCloud

```objc
AVObject *todo = [[AVObject alloc] initWithClassName:@"Todo"];
[todo setObject:@"做完PPT" forKey:@"content"];
[todo save];
```

网站控制台


数据
Class Name
_Conversation 1.2k
_File 0
_Followee 0
_Follower 0
_Installation 40
_Role 0
_User 0
Todo 1
LeanCloud 从 9 月 18 日开启余额检查和欠费自动停服机制：因为中秋假期，原定从 9 月 15 日起开启余额
足，服务将自动停止。详情请阅读：https://blog.leancloud.cn/5060/
添加行
删除行
添加列
查询
刷新
其他
编辑单元格
objectId STRING
content STRING
createdAt DATE
581e2b6a2f301e005c182be2
做完PPT
2016-11-06 02:56:42

如何确保代码永远正确?

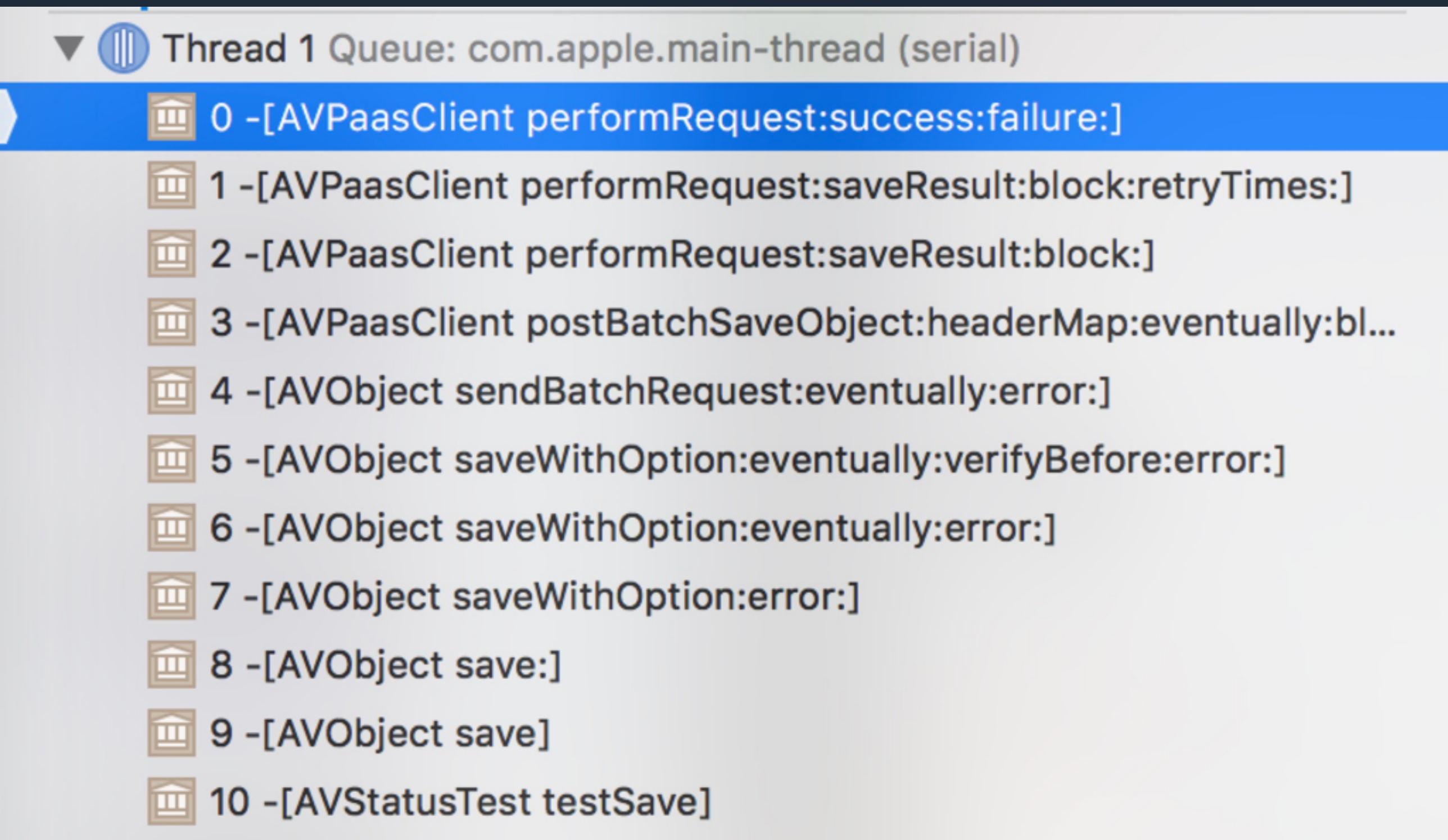
从 **LeanCloud** 获取该对象，看是否存在且一致

完整的测试代码

```
699 - (void)testSave {
700
701     AVObject *todo = [[AVObject alloc] initWithClassName:@"Todo"];
702     [todo setObject:@"做完PPT" forKey:@"content"];
703     BOOL saved = [todo save];
704     XCTAssertTrue(saved);
705
706     AVObject *fetchedTodo = [AVObject objectWithClassName:@"Todo" objectId:todo.
            objectId];
707     [fetchedTodo fetch];
708     XCTAssertEqualObjects([fetchedTodo objectForKey:@"content"], @"做完PPT");
709
710 }
```

这段测试发生了什么？

到 [dataTask resume] 发网络请求时经历了这些层级调用



涉及到的逻辑

- 更改了哪些字段，生成请求内容的逻辑
- 缓存逻辑
- 批量请求的逻辑
- **POST** 请求逻辑
- 每层的错误处理

几行测试代码确保了内部的成百上千行代码运行基本正确

10秒的时间就能验证原先需要3分钟验证的代码

测试来代替手工验证

- 我改了 **save** 里的代码，到底有没有改错呢？现在还能正确执行吗？还能保存到服务器吗？

- 跑一遍测试

对于复杂的情况，测试可以代替繁琐的手工验证

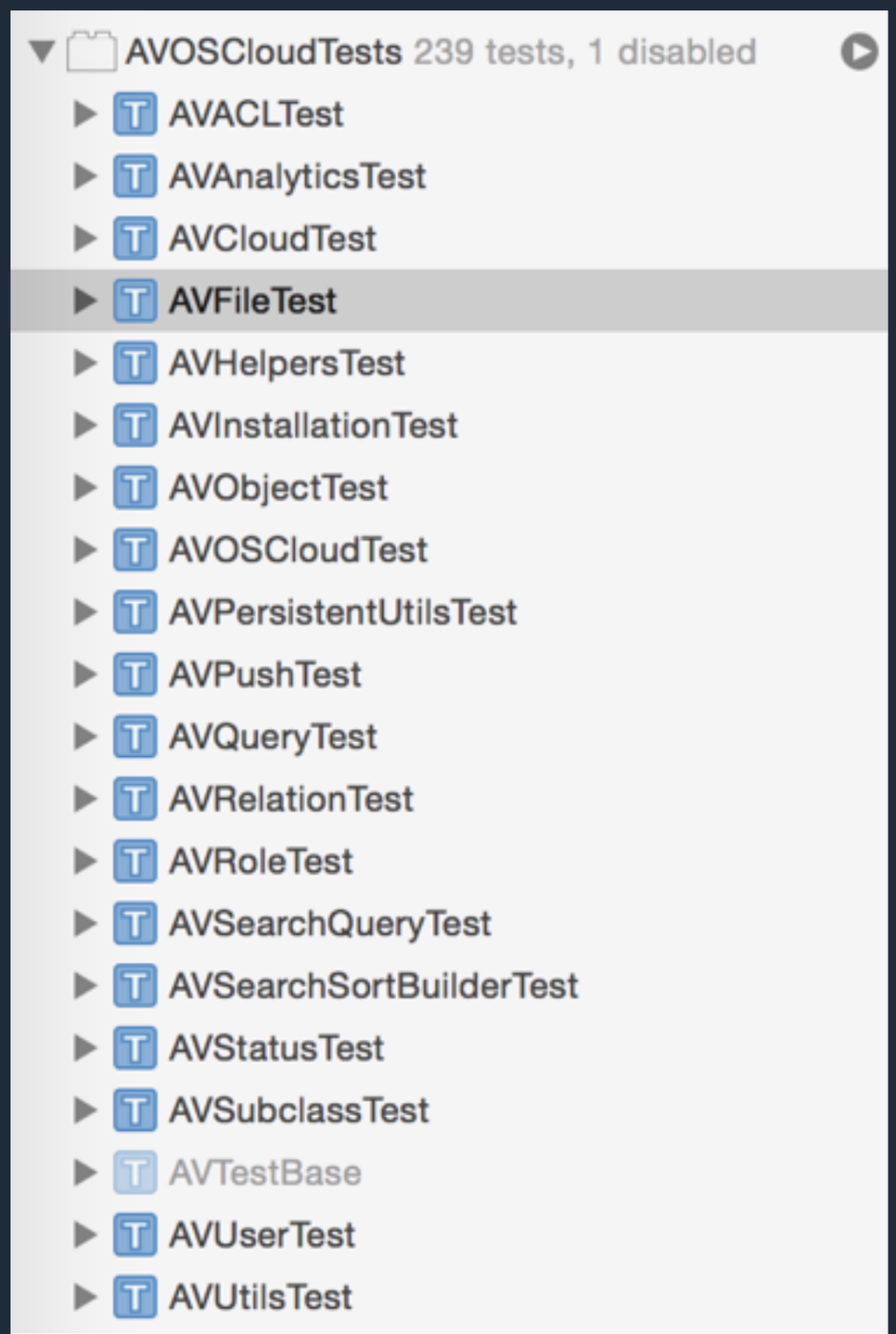
测试的本质

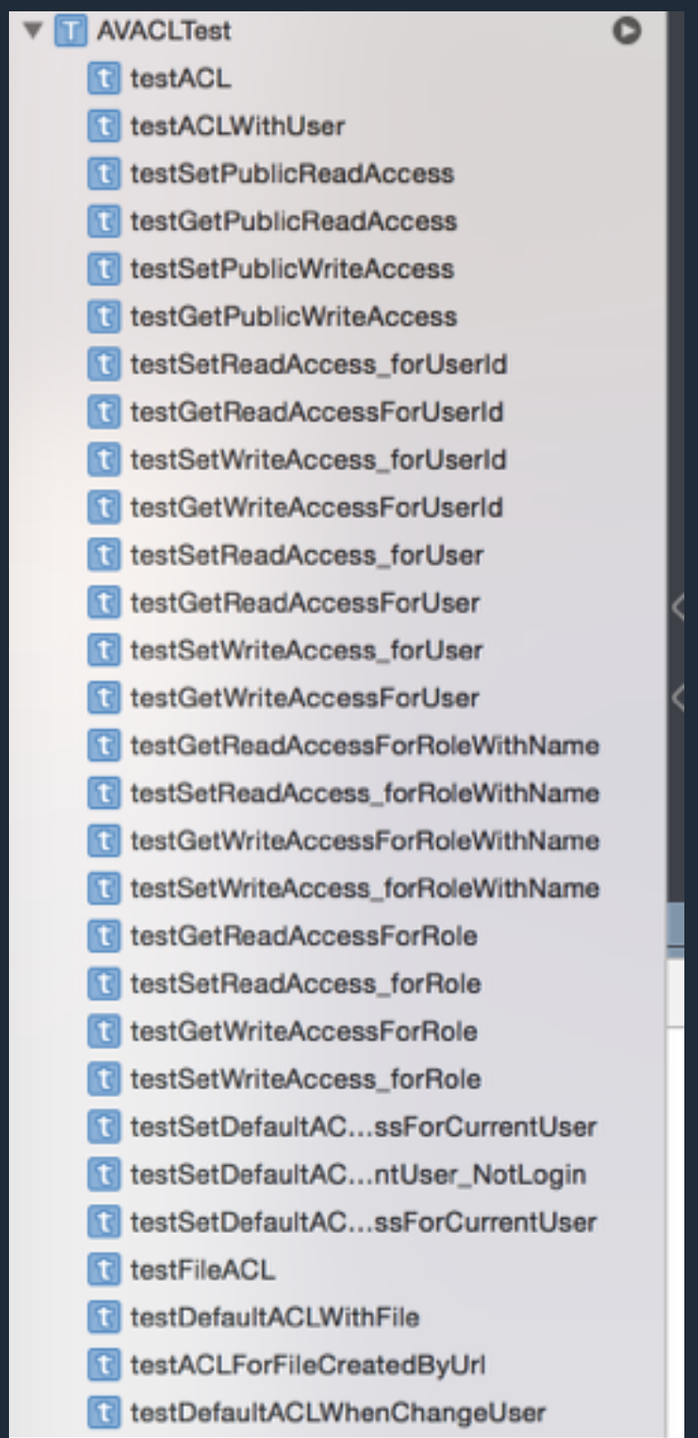
- 让强大的计算机来帮助你低效的手工验证

从此可以放心改 save 里的代码

- 如果改错了，比如没有保存到服务器，那么会获取对象失败
- 改完内部逻辑之后，跑一遍单元测试，能放心当前的改动没有影响旧的测试
- 如果在其他地方不小心

LeanCloud iOS SDK





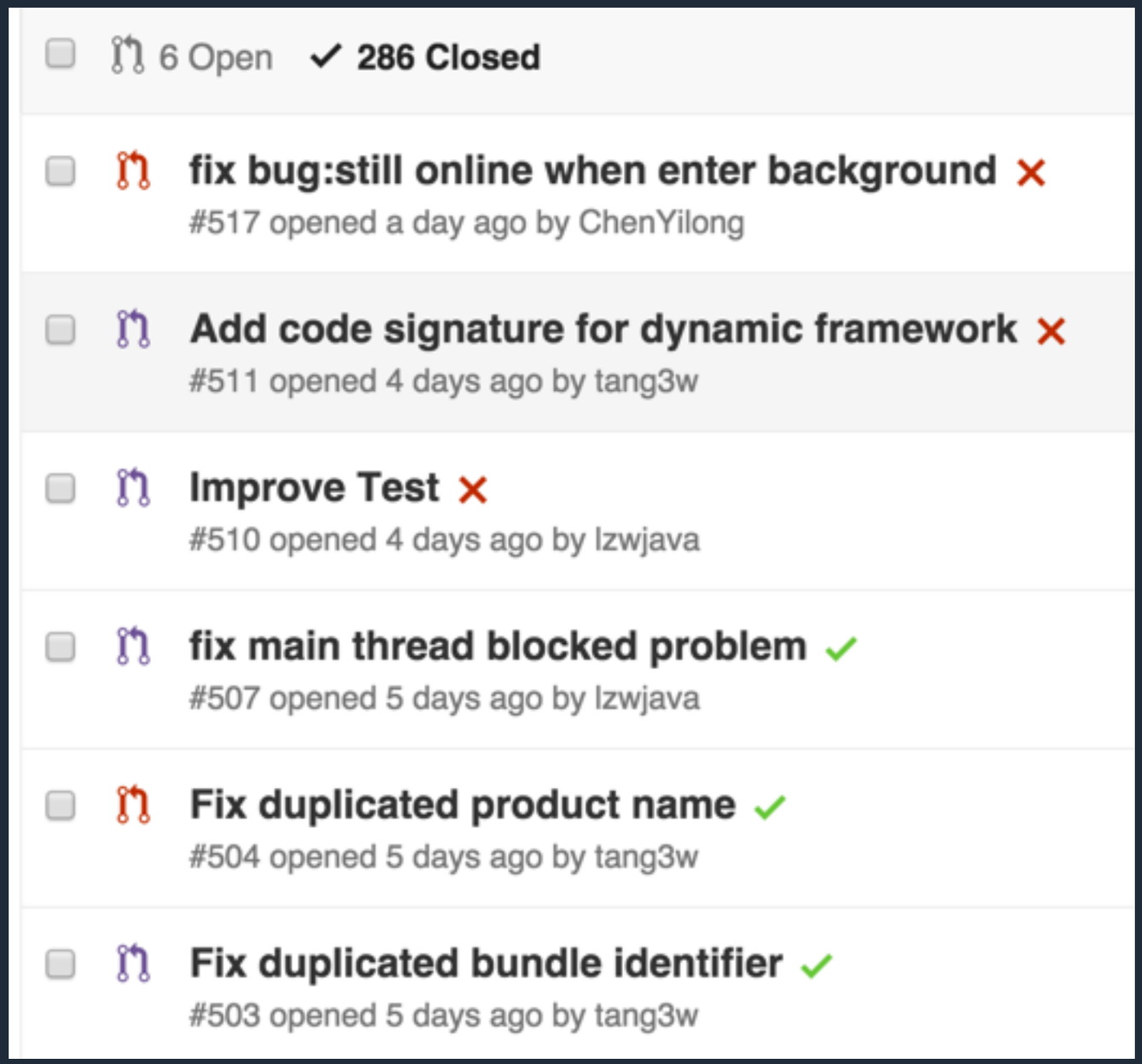
LeanCloud 的测试情况（一）

- 存储框架，数据层，适合写单元测试
- 项目早期就有单元测试
- 提交一个 **PR**，**Jenkins** 上跑单元测试
- 覆盖率约 **80%**

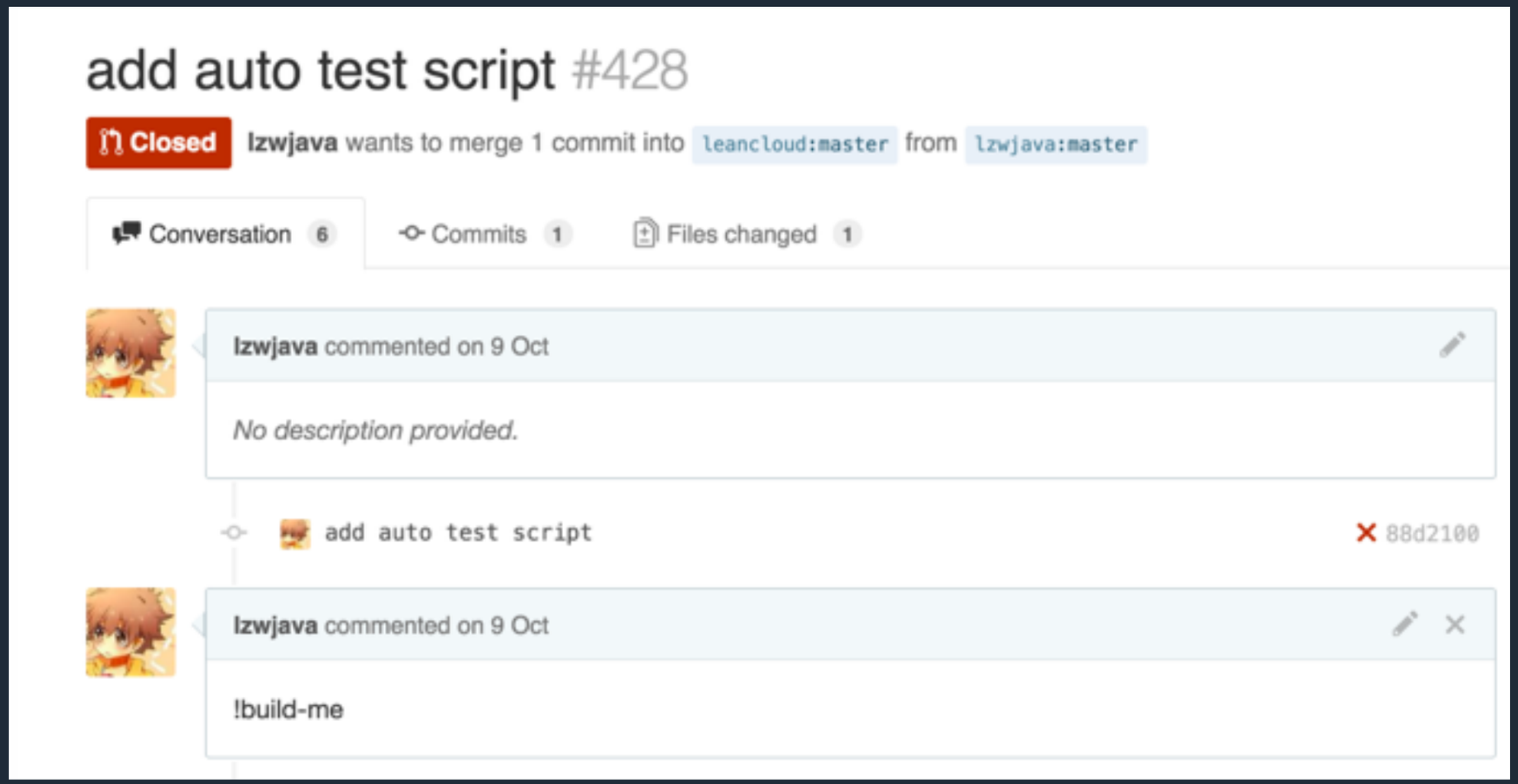
LeanCloud 的测试情况(二)

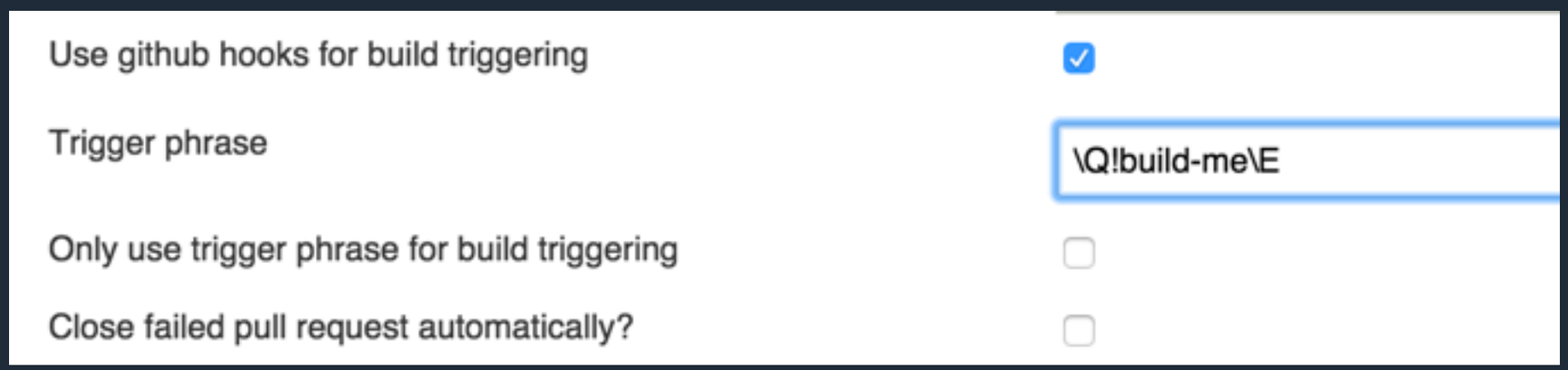
- 1）直接写测试，来检验正在开发的新接口。2）直接写测试，来重现 Bug 。
- 积累的测试越多越强大。
- 两种场景：改完一些代码后，本地跑相关测试，发现某些不通过；Jenkins 自动化跑测试的时候，发现最新改动引起其它问题，某些测试不通过
- 放心地发布与重构，省下手动验证的时间

提交 PR 跑测试


6 Open
286 Closed
fix bug:still online when enter background
#517 opened a day ago by ChenYilong
Add code signature for dynamic framework
#511 opened 4 days ago by tang3w
Improve Test
#510 opened 4 days ago by lzwjava
fix main thread blocked problem
#507 opened 5 days ago by lzwjava
Fix duplicated product name
#504 opened 5 days ago by tang3w
Fix duplicated bundle identifier
#503 opened 5 days ago by tang3w

输入暗号触发测试


add auto test script #428
Closed
lzwjava wants to merge 1 commit into leancloud:master from lzwjava:master
Conversation 6
Commits 1
Files changed 1
lzwjava commented on 9 Oct
No description provided.
add auto test script
88d2100
lzwjava commented on 9 Oct
!build-me


Use github hooks for build triggering
Trigger phrase
\Q!build-me\E
Only use trigger phrase for build triggering
Close failed pull request automatically?

Test Flow 1

```
1184 - (void)testSaveDescriptionKey {
1185     NSString * className = NSStringFromClass([self class]);
1186     NSError * error = nil;
1187     AVObject * object = [AVObject objectWithClassName:className];
1188     [object setObject:@"test" forKey:@"description"];
1189     [object save:&error];
1190     XCTAssertNil(error, @"%@", error);
1191     AVQuery* query = [AVQuery queryWithClassName:className];
1192     [query getObjectInBackgroundWithId:object.objectId block:^(AVObject *model, NSError *error) {
1193         XCTAssertNil(error, @"%@", error);
1194         XCTAssertEqualObjects([model objectForKey:@"description"], @"test");
1195         NOTIFY
1196     }];
1197     WAIT
1198 }
```

用户反应说保存有 description 为 key 的对象时会报错，我写了测试来验证，接着找到问题，修复后，保留下了测试。

Test Flow 2

```
1166 - (void)testSetNull {
1167     AVObject *object = [self objectForAVObjectTest];
1168     [object setObject:[NSNull null] forKey:@"name"];
1169     NSError *error;
1170     [object save:&error];
1171 
1172     [AVObject setConvertingNullToNil:NO];
1173     AVObject *fetchedObject = [self fetchObjectById:object.objectId];
1174     NSString *name = [fetchedObject objectForKey:@"name"];
1175     XCTAssertNotNil(name);
1176     XCTAssertTrue([name isEqual:[NSNull null]]);
1177 
1178     [AVObject setConvertingNullToNil:YES];
1179     NSString *name1 = [fetchedObject objectForKey:@"name"];
1180     XCTAssertNil(name1);
1181     XCTAssertFalse([name1 isEqual:[NSNull null]]);
1182 }
```

开发新接口时，写了相应的代码后，我加入了测试，
看看刚写的代码是否工作正常。

Test Flow 3

- 改了 **AVObject.m** 的代码后，我运行 **AVObjectTest.m** 测试，看看改动是否造成测试不通过

Test Flow 4

- 提交 PR，Jenkins 自动跑测试

总结：为什么要写单元测试

- 当发现自己改了代码后，经常需要手动验证是否正常时
- 发现改了代码后，经常引起其它地方的 **Bug** 时
- 多人合作的项目、可能交给其他人的项目
- 高质量、流行的开源项目一般都有丰富的单元测试

如何写单元测试

- 模块化代码，数据层和 **UI** 层分离
- 最少的测试代码达到最高的覆盖率
- 异步处理
- 框架选择
- 覆盖率

框架评析

- **expecta　　expect(error).not.beNil()**
- **specta　　describe("") it("")**
- **Kiwi　　describe("") it("")**
- **TDD** 和 **BDD** 框架的坏处：和 **Xcode** 结合不好，比如没有测试按钮，左边栏没有列全单元测试。

异步处理

```
130 - (void)waitNotification:(const void *)notification {
131     NSString *name = [NSString stringWithFormat:@"%p", notification];
132     [self expectationForNotification:name object:nil handler:nil];
133     [self waitForExpectationsWithTimeout:AV_YEAR_SECONDS handler:nil];
134 }
135
136 - (void)postNotification:(const void *)notification {
137     NSString *name = [NSString stringWithFormat:@"%p", notification];
138     [[NSNotificationCenter defaultCenter] postNotificationName:name object:nil];
139 }
```

```
1 #define WAIT                                                            \
2 do {                                                                    \
3 [self expectationForNotification:@"LCUnitTest" object:nil handler:nil]; \
4 [self waitForExpectationsWithTimeout:60 handler:nil];                  \
5 } while(0);
6
7 #define NOTIFY
      \
8 do {
      \
9 [[NSNotificationCenter defaultCenter] postNotificationName:@"LCUnitTest" object:nil];
      \
0 } while(0);
1
```

最终版本

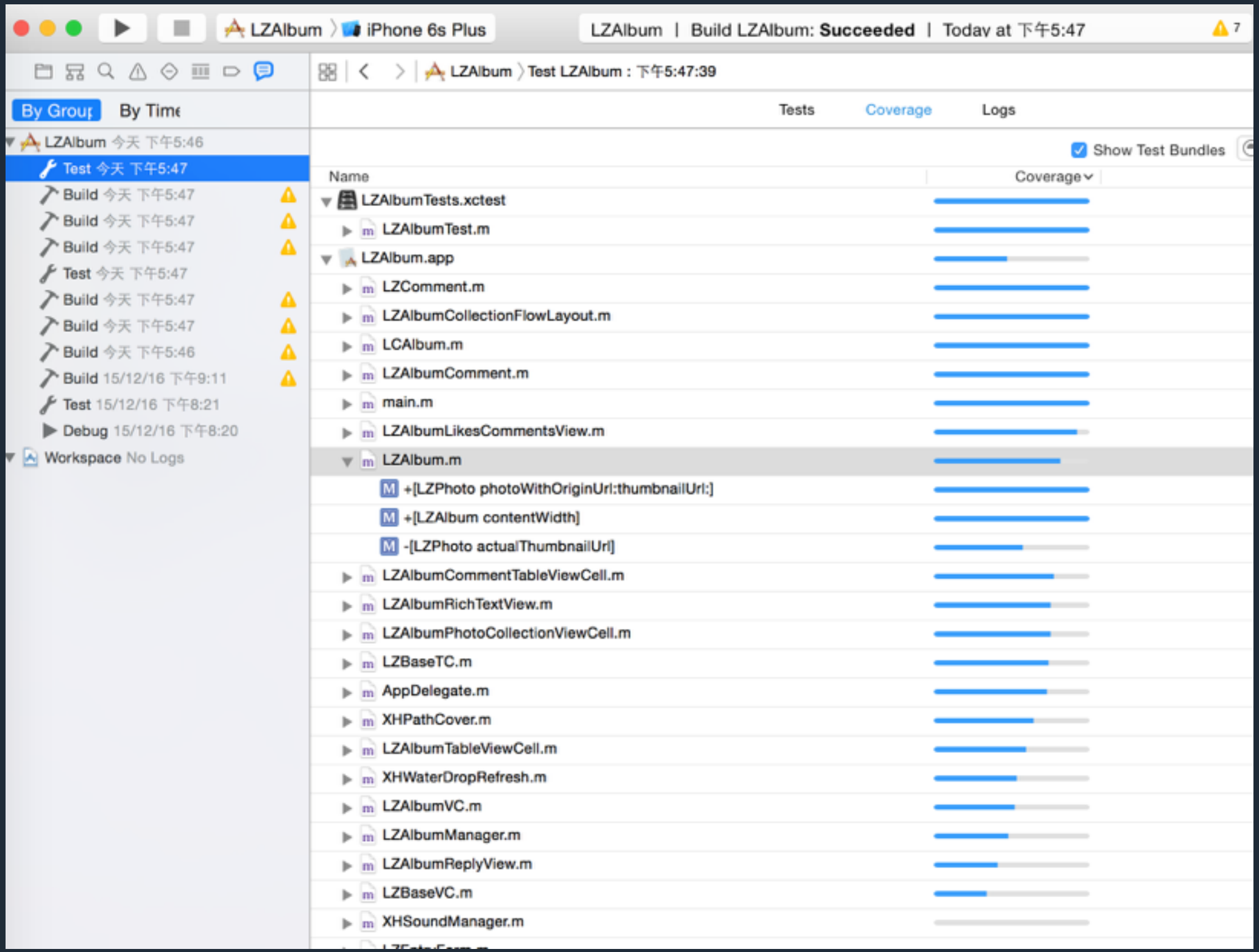
```
#define assertTrue(expr)                XCTAssertTrue((expr), @"")
#define assertFalse(expr)               XCTAssertFalse((expr), @"")
#define assertNil(a1)                   XCTAssertNil((a1), @"")
#define assertNotNil(a1)                XCTAssertNotNil((a1), @"")
#define assertEqual(a1, a2)             XCTAssertEqual((a1), (a2), @"")
#define assertEqualObjects(a1, a2)      XCTAssertEqualObjects((a1), (a2), @"")
#define assertNotEqual(a1, a2)          XCTAssertNotEqual((a1), (a2), @"")
#define assertNotEqualObjects(a1, a2)   XCTAssertNotEqualObjects((a1), (a2), @"")
#define assertAccuracy(a1, a2, acc)     XCTAssertEqualWithAccuracy((a1),(a2),(acc))

#define WAIT                                                                \
do {                                                                        \
[self expectationForNotification:@"LCUnitTest" object:nil handler:nil]; \
[self waitForExpectationsWithTimeout:60 handler:nil];                       \
} while(0);

#define NOTIFY                                                              \
do {                                                                        \
[[NSNotificationCenter defaultCenter] postNotificationName:@"LCUnitTest" object:nil];  \
} while(0);
```

```
- (void)testFindAlbums {
    [[LZAlbumManager manager] findAlbumWithBlock:^(NSArray *objects, NSError *error)
        {
        assertNil(error);
        assertTrue(objects.count > 0);
        NOTIFY
    }];
    WAIT
}
```

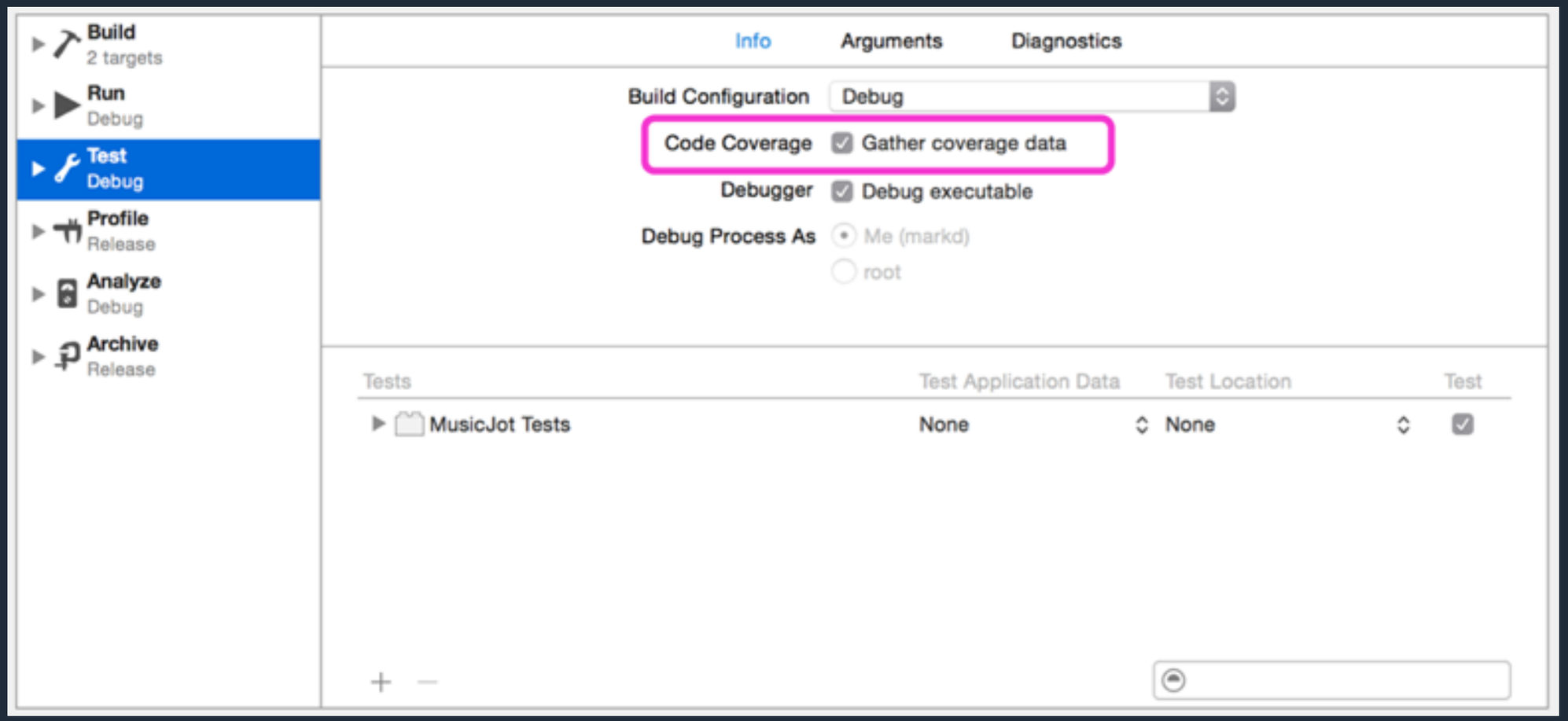
覆盖率


LZAlbum 〉 iPhone 6s Plus
LZAlbum | Build LZAlbum: Succeeded | Today at 下午5:47
7
LZAlbum 〉 Test LZAlbum : 下午5:47:39
By Group
By Time
Tests
Coverage
Logs
Show Test Bundles
Name
Coverage
LZAlbum 今天 下午5:46
Test 今天 下午5:47
Build 今天 下午5:47
Build 今天 下午5:47
Build 今天 下午5:47
Test 今天 下午5:47
Build 今天 下午5:47
Build 今天 下午5:47
Build 今天 下午5:46
Build 15/12/16 下午9:11
Test 15/12/16 下午8:21
Debug 15/12/16 下午8:20
Workspace No Logs
LZAlbumTests.xctest
LZAlbumTest.m
LZAlbum.app
LZComment.m
LZAlbumCollectionFlowLayout.m
LCAlbum.m
LZAlbumComment.m
main.m
LZAlbumLikesCommentsView.m
LZAlbum.m
+[LZPhoto photoWithOriginUrl:thumbnailUrl:]
+[LZAlbum contentWidth]
-[LZPhoto actualThumbnailUrl]
LZAlbumCommentTableViewCell.m
LZAlbumRichTextView.m
LZAlbumPhotoCollectionViewCell.m
LZBaseTC.m
AppDelegate.m
XHPathCover.m
LZAlbumTableViewCell.m
XHWaterDropRefresh.m
LZAlbumVC.m
LZAlbumManager.m
LZAlbumReplyView.m
LZBaseVC.m
XHSoundManager.m

覆盖率报告

- **Xcode 7 开始，自带覆盖率报告功能**
- **设置要点：1) Scheme 中开启 Gather Coverage Data 2)针对于 App Target 来测试，而非 Test Target**
- **https://www.bignerdranch.com/blog/weve-got-you-covered/**

Gather Coverage Data



远程自动化测试

- 搭建 **Jenkins**
- 本地空闲的 **Mac** 电脑或将 **Mac** 电脑拿到数据中心
- 测试脚本，设置项目
- **Github PR** 构建插件，**https://github.com/janinko/ghprb**
- 配置 **Jenkins Job**
- 本地测试能发挥出测试的 **70%** 好处，远程自动化则 **100%**

配置 Jenkins

- 要让 **Jenkins** 能读取你的代码
- 配置 **Hook**，当提交 **GitHub PR** 时，发送相应的事件到 **Jenkins** 上
- **Jenkins** 要知道是哪一个 **PR**，来拉取最新代码
- 设置 **Jenkins** 测试脚本
- 当测试失败之后，要让 **Jenkins** 通知 **GitHub** 页面
- 测试失败后，可发送通知到 **Slack** 或邮件

Jenkins 如何知道测试失败或成功

- 根据进程返回值
- 如果是 **0**，则 **Jenkins** 认为测试成功，否则失败
- **Xcode 7** 之前，尽管 **xcodebuild test** 失败了，依然返回**0**；**Xcode 7** 之后，失败了则返回 **65**

test-all.sh

```
#!/bin/bash -v

xcodebuild clean                        \
    -workspace AVOS.xcworkspace         \
    -scheme AVOSCloudIMTests            \

script -q -t 0 test.log                 \
    xcodebuild                          \
        -workspace AVOS.xcworkspace     \
        -scheme AVOSCloudIMTests        \
        -configuration Release          \
        -sdk iphonesimulator            \
        test

grep "\*\* TEST FAILED \*\*" test.log  > /dev/null
if [[ $? -eq 0 ]]; then
    rm test.log
    exit 1
fi

xcodebuild clean                        \
    -workspace AVOS.xcworkspace         \
    -scheme AVOSCloudTests              \

script -q -t 0 test.log                 \
    xcodebuild                          \
        -workspace AVOS.xcworkspace     \
        -scheme AVOSCloudTests          \
        -configuration Release          \
        -sdk iphonesimulator            \
        test

grep "\*\* TEST FAILED \*\*" test.log  > /dev/null
if [[ $? -eq 0 ]]; then
    rm test.log
    exit 1
else
    rm test.log
    exit 0
fi
```

Xcode 7之前的测试脚本

1. 检测是否有 TEST FAILED 字符串

Xcode 7 之后

test-all.sh

```
#!/bin/bash -v

xcodebuild                                \
    -workspace AVOS.xcworkspace \
    -scheme AVOSCloudIMTests    \
    -sdk iphonesimulator        \
    clean                       \
    test    &&                  \
xcodebuild                      \
    -workspace AVOS.xcworkspace \
    -scheme AVOSCloudTests      \
    -sdk iphonesimulator        \
    clean                       \
    test

```

Travis CI

- 设置 **Test Scheme** 为 **shared**
- **Travis** 上连接 **GitHub** 账号，读取项目
- **.travis.yml**
- **Travis** 免费版可以给开源项目测试、收费版给私有项目测试
- **Travis** 测试比较慢
- 如果会用 **Jenkins** 自己搭自动化测试框架，**Travis** 眨眼间就会

远程打包发布

- 发布脚本
- **Jenkins** 上配置 **Job** 运行
- **LeanCloud** 上是发布框架，远程打包 **SDK** 推送到 **GitHub** 仓库，同步网站更新
- 针对于 **App**，似乎没必要远程打包发布

实现思路

- 让 **Jenkins** 读取你的代码
- 读取发布版本
- **unlock keychain**，构建一个项目会弹出输入密码框，来读取签名的 **keychain**

unlock-keychain

```
# Unlock the build certificate
execute_command('security unlock-keychain -p "xxx" ~/Library/Keychains/login.keychain')
```

在命令行中解 **Keychain**。一般会弹框输入密码

LeanCloud iOS SDK

- https://github.com/leancloud/objc-sdk
- 自动化测试
- 自动化打包 **SDK**
- 自动化生成、推送 **podspec**
- 自动化发布

脚本:命令行build

```ruby
def build_target(target_name, sdk, build_dir, archs, misc='')
  command = <<-EOC.strip.gsub(/^[ \t]+/, '')
  xcodebuild \
  -project #{project_path} \
  -target #{target_name} \
  -configuration #{configuration}
  EOC

  command += " -sdk #{sdk}" unless empty_string?(sdk)
  command += " CONFIGURATION_BUILD_DIR=\"#{build_dir}\"" unless empty_string?(build_dir)
  command += " ARCHS=\"#{archs}\"" unless empty_string?(archs)
  command += " #{misc}" unless empty_string?(misc)

  execmd command
end
```

脚本：命令行build

```ruby
def build_ios_sdk(target_name)
  clean_target target_name

  prefix = File.join(slice_build_path, target_name)

  build_target target_name, 'iphonesimulator', File.join(prefix, 'iphonesimulator'), 'i386 x86_64
  build_target target_name, 'iphoneos',        File.join(prefix, 'iphoneos'),        'armv7 armv7

  merge_frameworks(prefix, target_name)
end
```

脚本：合并framework

```
def merge_frameworks(prefix, target_name)
  product_name = product_name(target_name)

  frameworks = Dir.glob(File.join(prefix, '**', "#{product_name}.framework"))
  first_framework = frameworks.first

  return if first_framework.nil?

  output = File.join(build_path, "#{target_name}.framework")
  execmd "cp -RLp #{first_framework} #{output}"

  return if frameworks.count == 1

  executables  = frameworks.map { |framework| File.join(framework, product_name) }

  lipo File.join(output, product_name), *executables
end
```

脚本：拉取远程代码

```
remote_url = 'git@github.com:leancloud/objc-sdk.git'

tags = execute_command "git ls-remote --tags #{remote_url}"
abort 'Git tag not found on remote repository. You can push one.' unless tags.include?

commit_sha = tags[/([0-9a-f]+)\srefs\/tags\/#{version}/, 1]

temp_remote = "_origin-temp-remote-for-deployment"
temp_branch = "_branch-temp-branch-for-deployment"

execute_command "git remote remove #{temp_remote} >/dev/null 2>&1", false
execute_command "git remote add #{temp_remote} #{remote_url} >/dev/null 2>&1", false
execute_command "git fetch #{temp_remote} --tags >/dev/null 2>&1"
execute_command "git checkout -b #{temp_branch} #{commit_sha} >/dev/null 2>&1"
```

脚本：生成 Podspec

```
template = File.read('Podspec/AVOSCloud.podspec.mustache')

podspec = Mustache.render template, {
    'version'                => version,
    'source_files'           => "'AVOS/AVOSCloud/**/*.{h,m}'",
    'resources'              => "'AVOS/AVOSCloud/AVOSCloud_Art.inc'",
    'public_header_files'    => file_list_string(public_header_files, 4),
    'osx_exclude_files'      => file_list_string(osx_exclude_files, 4),
    'watchos_exclude_files' => file_list_string(watchos_exclude_files, 4),
    'xcconfig'               => "{'OTHER_LDFLAGS' => '-ObjC'}"
}

write 'AVOSCloud.podspec', podspec
```

脚本：自动化推送 Podspec

```ruby
def push_podspec_in_path(path)
  iterate_r(path, 'podspec') do |file|
    podspec_name = File.basename(file, ".podspec")
    podspec_version = version[1..-1]
    exists = is_podspec_exists(podspec_name, podspec_version)
    if exists
      log("#{podspec_name} #{podspec_version} exists!")
      next
    else
      log("#{podspec_name} #{podspec_version} not exists, now try to push it")
    end

    ok = false
    for i in 0..10
      ok = system("pod trunk push #{file} --verbose --allow-warnings")
      if ok
        log("succeed to push #{file}")
        break
      else
        log("failed to push #{file}")
      end
    end

    if !ok
      exit_with_info('fail to push podspec, please check.')
    end
  end
end
```

重复推送 Podspec

```ruby
def is_podspec_exists(name, version)
  podspec_url = "https://github.com/CocoaPods/Specs/blob/master/Specs/#{name}/#{version}/#{name}
  url = URI.parse(podspec_url)
  req = Net::HTTP::Get.new(url.to_s)
  http = Net::HTTP.new(url.host, url.port)
  http.use_ssl = true
  res = http.request(req)
  res.code == '200'
end
```

lipo

- 查看 **framework** 的编译架构 **lipo -info**
- **lipo** 合并不同架构的 **framework**

代码质量

代码质量

- 一致性
- 不写重复代码
- 少用全局变量
- 逻辑分层
- 每一行代码都应有它存在的意义

零碎的知识

- **lipo** 使用
- 清空所有的生成文件，**Clean Build Folder**
- **Xcode** 快捷键，根据当前文件展开左侧导航、**Open Quickly**、查看 **Macro** 预编译、**.h** 与 **.m** 文件跳转
- **Pod** 高级用法
- 如何制作 **Framework**
- **Xcode Configurations**
- **Reveal In GitHub** 插件
- **Instrument** 工具使用，定位代码

Xcode 快捷键

- **Shift + Command+ J**
- **Shift + Command + Up/Down**
- **Shift + Command + O**
- **Emacs Way**
- 根据语法来删除，**Delete expression backward**
- **preferences Xcode** 设置中，搜索快捷键的 **keyword**

Delete expression backward

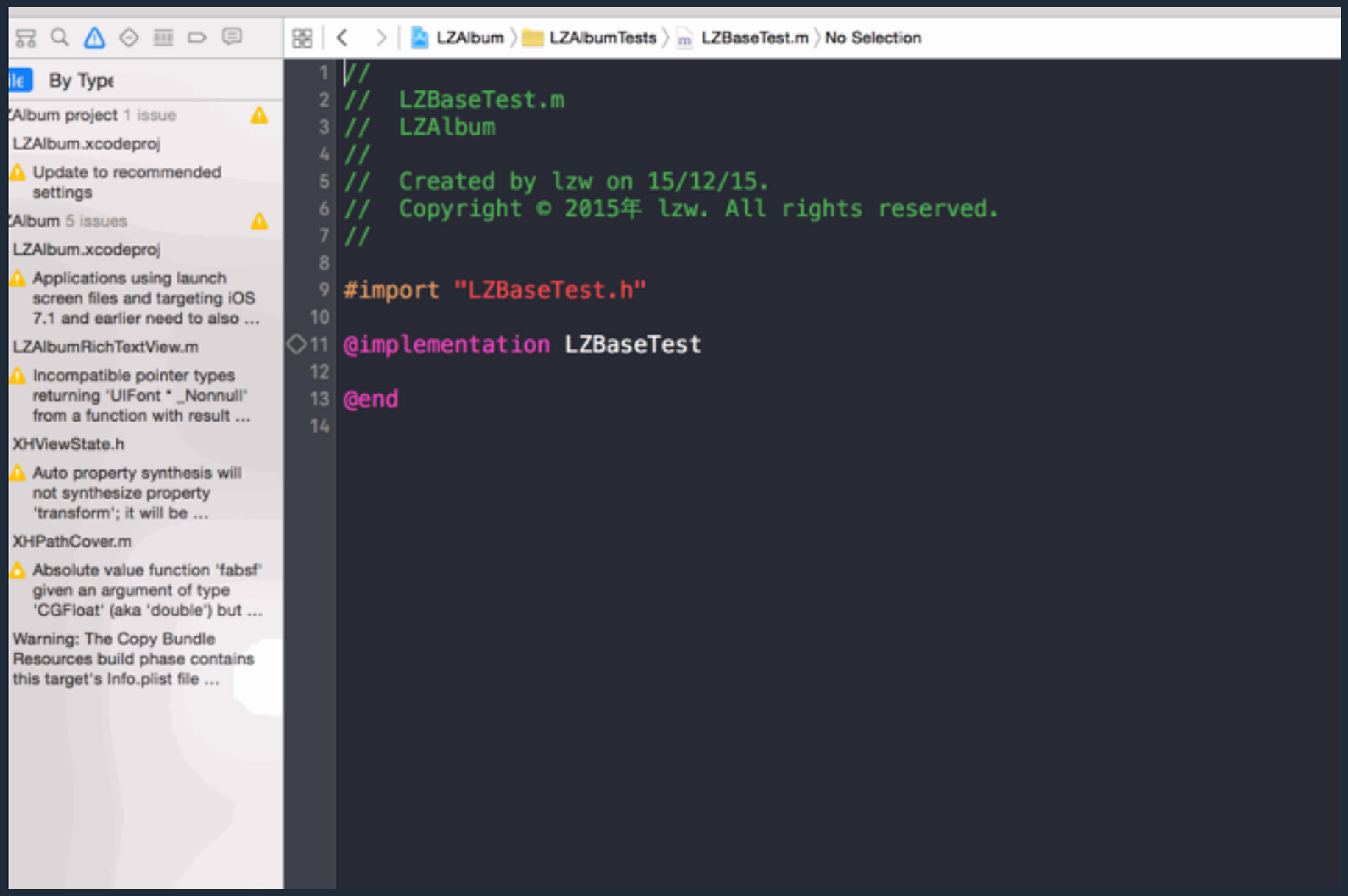
```
62 - (void)testSaveDescriptionKey {
63     NSString * className = NSStringF
64     NSError * error = nil;
65     AVObject * object = [AVObject ob
66     [object setObject:@"test" forKey
67     [object save:&error];
68     XCTAssertNil(error, @"%@", error
69     AVQuery* query = [AVQuery queryW
70     [query getObjectInBackgroundWith
           error) {
71         XCTAssertNil(error, @"%@", e
72         XCTAssertEqualObjects([model
73         NOTIFY
74     }];
75     WAIT
76 }
```

```
2 - (void)testSaveDescriptionKey
```

Pod

- 本地引用 **pod**，好处
- **—no-repo-update —verbose**
- 发布 **pod**、**podfile** 配置

Clean Build Folder

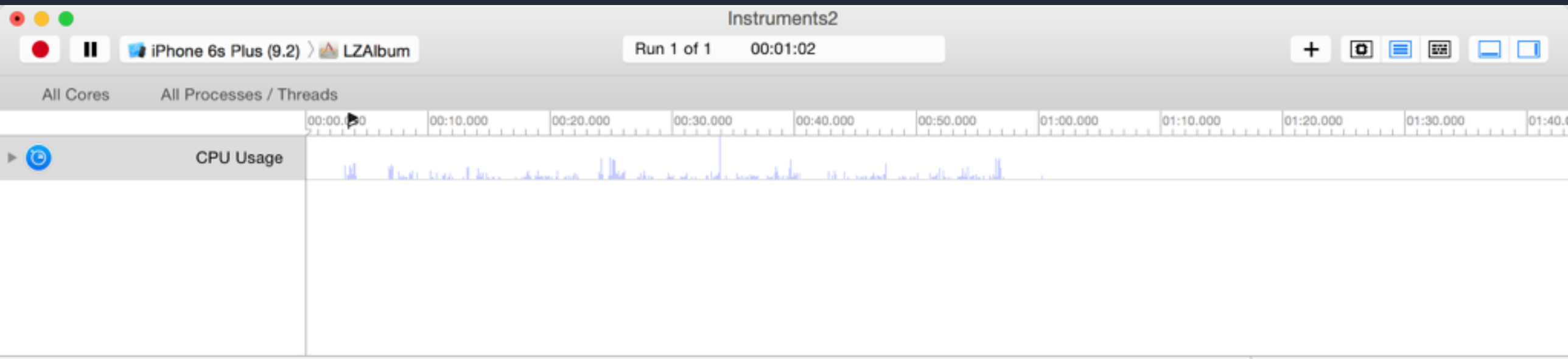


1）Alt 键触发 2）两小时捣鼓不如 Clean Build Folder 一下
3）Duplicate Symbol 等 Build 的问题时很好用

如何制作 Framework

- 动态库和静态库的区别
- 静态库如何用动态库伪装，包含 .a 和 headers
- 打包模拟器和真机都可以使用的 Framework

Instruments



好用工具介绍

- **Reveal +** 越狱 **iPhone**，分析任意 **App** 的 **UI** 界面
- **Flex +** 越狱 **iPhone**，分析任意 **App** 的网络请求、**UI**界面、本地文件、**NSUserDefaults**、**Log** 等

Flex 1


无 SIM 卡
22:47
87%
醉在花香里的...
日揭晓，@李鍾碩 @贾乃亮 @钟汉良 @
menu
views
select
move
close
希望谁获奖？10月19日—29日为TA献
花，“网络人气男神”投票截止11月2
日！快点抓紧时间！获奖者还将有神秘
惊喜！进入专题区投票去>>我正在看
专题：《2015亚洲影响力东方盛典》
21
首页
2
消息
发现
我


无 SIM 卡
22:47
87%
醉在花香里的...
热门
menu
views
select
move
close
UIImageView {(0, 7), (30, 30)}
#2015亚洲影响力东方盛典#将在11月3
日揭晓，@李鍾碩 @贾乃亮 @钟汉良 @
井柏然 @胡歌 @刘烨 @华晨宇yu @李易
峰 @Mr_凡先生 等提名候选人中，你最
希望谁获奖？10月19日—29日为TA献
花，“网络人气男神”投票截止11月2
日！快点抓紧时间！获奖者还将有神秘
惊喜！进入专题区投票去>>我正在看
专题：《2015亚洲影响力东方盛典》
21
首页
2
消息
发现
我

Flex 2


无 SIM 卡
22:48
87%
返回
Settings
搜索
15 Requests (162 KB received)
unread_count?skin=default&c=iphone&fr…
api.weibo.cn/2/remind
22:47:47 · GET · 200 OK · 997 字节 · 243ms…
unread_count?skin=default&c=iphone&fr…
api.weibo.cn/2/remind
22:47:17 · GET · 200 OK · 997 字节 · 163ms…
7dced06cjw1exxzga1ueqg20b40avqva.gif
ww4.sinaimg.cn/square
22:47:07 · GET · 200 OK · 5 KB · 360ms (29…
dcd0d42dgw1exzpk67ff7j20g80r5win.jpg
ww3.sinaimg.cn/square
22:47:07 · GET · 200 OK · 4 KB · 459ms (42…
6a165c19jw1ey1ejmw7enj20h60t8q8a.jpg
ww3.sinaimg.cn/square
22:47:07 · GET · 200 OK · 3 KB · 209ms (15…
activities?sdk=ios&advertiser_tracking_e…


无 SIM 卡
22:49
88%
返回
搜索
7 FILES (104.6 MB)
Documents
17 files
Library
9 files
StoreKit
1 file
Weibo.app
2395 files
iTunesArtwork
42 KB - 2015-01-17 15:56:40 +0000
iTunesMetadata.plist
2 KB - 2015-01-17 15:56:40 +0000
tmp
0 files

Thank you!

Thanks to 彦祖、
ufosky、tang3w、sunng87、iOS 程序猿、冉神
Thanks to LeanCloud CTO 允许我讲内部测试流程

微信

lzwjava