

# 目錄

Introduction	1.1
Notes	1.2
AppDelegate - iOS APP Launch Options	1.2.1
Auto Layout - 手动添加 Auto Layout 约束(Masonry)	1.2.2
Auto Layout - 约束冲突断点调试方法	1.2.3
Block - 介绍与使用	1.2.4
CoreData - 使用 FYHDBManager 管理 CoreData	1.2.5
Git - iOS 开发 gitignore 文件	1.2.6
Git - 如何 clone git 项目到一个非空目录	1.2.7
Git - 常见错误与解决方案	1.2.8
Git - 恢复被误删除分支	1.2.9
JSON - 使用 JSONHelper 进行 JSON 数据解析	1.2.10
Keychain - 使用 PDKeychainBindingsController 实现 Keychain 保存数据封装	1.2.11
KVO - 使用 KVO 更新 UITableViewCell 显示	1.2.12
Mac 系统 - Mac OS 运行shell脚本	1.2.13
Mac 系统 - 使用 Automator 批量重命名	1.2.14
Mac 系统 - 使用 Scp 命令上传下载文件	1.2.15
Mac 系统 - 修改 Mac OS X root 密码	1.2.16
Mac 系统 - 启用定时自动执行任务	1.2.17
Mac 系统 - 通过 Shell 脚本自动记录 Terminal 活动	1.2.18
NSArray - 对自定义对象的数组进行排序	1.2.19
NSData - NSData 数据压缩与解压缩	1.2.20
NSDate - 时间日期相关操作方法总结	1.2.21
NSMutableSet - 在NSMutableSet中添加自定义对象时怎么保证不重复	
NSObject - 实现自定义对象 isEqual 方法	1.2.23
NSString - 删除 NSString 中前后空格或回车符	1.2.24

NSString - 删除 NSString 中特定字符	1.2.25
NSString - 汉字繁体简体相互转换的实现	1.2.26
NSString - 汉字转为拼音显示的实现	1.2.27
NSString - 相似度检测	1.2.28
NSString - 筛选出 NSString 中特定字符串	1.2.29
NSTimer - 解决 NSTimer 的循环引用问题导致的内存泄漏	1.2.30
NSURL - 特殊字符处理	1.2.31
openURL - iOS App 跳转 App Store 下载和 App Store 评论	1.2.32
Runtime - 使用 Aspects 实现 Method Swizzling 和 AOP 实践	1.2.33
UDID - 解决方案介绍与比较	1.2.34
UIButton - badge 显示的实现(使用 UIBarButtonItem-Badge)	1.2.35
UIButton - 使用 RNLoadingButton 实现等待按钮	1.2.36
UIButton - 避免多次重复点击	1.2.37
UIDevice - 通过 DeviceKit 更方便的获取硬件设备参数	1.2.38
UIEvent - iOS 事件拦截	1.2.39
UIImage - 获取 UIImage 大小与压缩 UIImage	1.2.40
UIImageView - 使用 LBBlurredImage 实现图像模糊效果	1.2.41
UILabel - UILabel 边缘黑线问题	1.2.42
UILabel - 悬浮美金单位效果的实现	1.2.43
UILabel - 显示多格式文本	1.2.44
UILabel - 显示换行的方法	1.2.45
UILabel - 计算文本高度	1.2.46
UILabel - 通过字符串长度计算显示框大小的方法	1.2.47
UILabel \ UITextField - 设置行间距等富文本属性	1.2.48
UINavigationBar - 自定义按钮和标题	1.2.49
UINavigationController - 使用 SlideNavigationController 实现侧滑需求	
UIPageControl - 翻页显示的实现	1.2.50
UIScrollView - UIScrollView 添加 UIButton 时不响应滑动操作	1.2.52
UIScrollView - 给 UIScrollView 添加 Autolayout 约束条件	1.2.53
UISearchBar - 通过 UISearchDisplayViewController 实现全屏搜索显示效果	1.2.54

UITableView - 使用 EGOResfreshTableHeaderView 实现下拉刷新	
UITableView	1.2.55
UITableView - 滑动加载性能优化总结	1.2.56
UITableView \ UIScrollView - 通过 contentInset 属性设置周边间距	
UITableViewCell - 动态修改 UITableViewCell 高度	1.2.58
UITableViewCell - 1.2.57	
UITextField - 使用备忘	1.2.59
UITextView - 添加内容后自动滚动到最后一行	1.2.60
UIView - iOS 完全复制一个 UIView	1.2.61
UIView - 代码实现截图功能	1.2.62
UIView - 代码设置 UIView 的阴影、圆角、边框效果	1.2.63
UIButton \ UIButton - 独占响应事件	1.2.64
UIViewController - 旋转问题 willRotateToInterfaceOrientation 方法无法正常调用	1.2.65
UIWebView - 使用 UIWebViewToFile 实现 UIWebView 内容转为 Image 或 PDF	1.2.66
Xcode - iOS 开发崩溃(Crash)调试	1.2.67
Xcode - Xcodeproj 介绍	1.2.68
Xcode - 为 Xcode 添加 Snippets	1.2.69
Xcode - 使用 BBUncrustifyPlugin-Xcode 插件实现代码格式化	1.2.70
Xcode - 使用 chisel 提高 LLDB 调试效率	1.2.71
Xcode - 使用 DWURecyclingAlert 进行 UITableView 和 UICollectionView 的绘图性能测试	1.2.72
Xcode - 使用 Instruments 的 Allocations 工具检测内存分配情况	1.2.73
Xcode - 使用 Instruments 的 Core Animation 工具进行 UIKit 性能调优	
Xcode - 使用 Jenkins 进行 App 自动化打包	1.2.74
Xcode - 1.2.75	
Xcode - 使用 LLDB 调试代码	1.2.76
Xcode - 在 Xcode 里添加和使用 Debug 宏定义	1.2.77
Xcode - 插件开发学习备忘	1.2.78
Xcode - 插件的安装与使用	1.2.79
Xcode - 系统警告处理( 清除和添加 )	1.2.80
Xcode - 调试相关	1.2.81

Xcode - 运行时环境变量( Environment Variables )	1.2.82
Xcode - 非 App Store 下载官方 Xcode 安装包方法	1.2.83
Xib - 通过 Xib 加载自定义 Cell	1.2.84
Xib - 通过 Xib 加载自定义 View	1.2.85
专题 - iOS 使用 XMPP 实现即时聊天	1.2.86
专题 - iOS 应用证书、真机调试与发布 Appstore	1.2.87
专题 - iOS 开发学习资源参考	1.2.88
专题 - iOS 开发工具软件集合	1.2.89
专题 - iOS 开发第三方代码库参考	1.2.90
专题 - iOS 开发调试代码集合	1.2.91
专题 - iOS 设备处理器指令集参考	1.2.92
专题 - Mac & iOS 完整开源项目源代码参考	1.2.93
专题 - objc.io Issue 3: 视图 相关文章五篇	1.2.94
专题 - 优秀技术博客参考	1.2.95
专题 - 优秀技术文章参考	1.2.96
专题 - 值得模块化的业务需求	1.2.97
专题 - 创建自己的 iOS 框架	1.2.98
专题 - 可视控件介绍与可定制替代方案推荐	1.2.99
专题 - 后台模式开发指南	1.2.100
专题 - 控制器介绍	1.2.101
专题 - 时间戳的介绍与使用	1.2.102
二维码 - QRCode 生成与识别	1.2.103
其他 - RESTful API 设计与最佳实践介绍	1.2.104
其他 - 书籍推荐	1.2.105
其他 - 有趣的网址收集	1.2.106
内购 - iOS 内购的快速实现	1.2.107
动画 - Core Animation 之 CABaseAnimation	1.2.108
动画 - Core Animation 之 CATransform3D	1.2.109
动画 - Core Animation 之 Key Path	1.2.110
动画 - 使用 POViewFrameBuilder 快速实现 UIView 的动画移动和布局	
	1.2.111

动画 - 页面跳转 - 自定义模态跳转动画	1.2.112
地图 - MKMapView 地图开发相关总结	1.2.113
地图 - 为 Xcode 工程添加自定义初始化位置 (载入 gpx 文件)	1.2.114
地图 - 使用 Eviltransform 进行火星坐标转换 (大陆地区地理坐标偏移问题)	1.2.115
字体 - 添加自定义字体	1.2.116
工具 - Markdown 简明教程	1.2.117
工具 - 使用 Cuterman 插件进行快速切图	1.2.118
工具 - 使用 iBackupbot 查看非越狱真机应用文件	1.2.119
工具 - 使用 objc_dep 检查项目中的导入依赖(Import Dependancies)	
工具 - 使用 VisualJSON 进行 JSON 格式网络接口 API 测试	1.2.120
工具 - 使用 Xcodebuild 进行命令行打包	1.2.122 1.2.121
应用间通信 - App 之间共享数据的几种方法	1.2.123
应用间通信 - URL Schemes	1.2.124
应用间通信 - 实现应用间互相调用与数据传递	1.2.125
应用间通信 - 文档导入导出实现	1.2.126
应用间通信 - 通过 URL 检测是否安装并打开应用	1.2.127
手势 - 判断点击位置是否在某区域 (View) 内	1.2.128
手势 - 实现手势操作介绍	1.2.129
授权 - 使用 ClusterPermissions 更加友好的提示授权操作	1.2.130
数据加密 - 3DES 加密(包含JAVA服务器端解密)	1.2.131
数据加密 - AES 加密解密	1.2.132
数据加密 - Base64 编码 ( NSData <=> NSString )	1.2.133
数据加密 - DES 加密(包含PHP服务器端解密)	1.2.134
数据加密 - MD5 加密	1.2.135
数据加密 - RSA 加密(包含JAVA服务器端解密)	1.2.136
数据加密 - 使用 NAVAJO 进行密码安全强度检测	1.2.137
数据存储 - 使用 SSZipArchive 实现文件的压缩和解压缩	1.2.138
数据存储 - 序列化对象	1.2.139
数据存储 - 文件(目录)操作笔记	1.2.140
数据库 - 使用 SQLCipher 进行数据库加密存储	1.2.141

文档 - iOS 人机交互指南	1.2.142
文档 - iOS 项目的目录结构	1.2.143
文档 - 苹果审核未通过问题与解决方案参考	1.2.144
时间 - 使用 Benchmarking 精确测量的代码运行时间	1.2.145
时间 - 倒计时器的实现	1.2.146
时间 - 延时执行解决方案	1.2.147
正则表达式 - 在 iOS 开发中使用正则表达式	1.2.148
测试 - iOS 应用测试 Checklist 以及思维导图	1.2.149
测试 - iOS 自动化测试框架介绍	1.2.150
测试 - 使用 DVR 进行网络请求数据的录制与测试	1.2.151
测试 - 异步函数的单元测试	1.2.152
版本兼容 - 7\8 - 无法正确获取 iPad 横竖屏宽高解决方案	1.2.153
特殊控件 - 使用 DBSphereTagCloud 实现标签云效果	1.2.154
特殊控件 - 使用 PSCyclingImageView 实现自动循环轮播图	1.2.155
特殊控件 - 使用 RecordingCircleOverlayView 实现环形记录仪动画	
特殊控件 - 使用 SubjectiveCPhotoPanner 实现动作感应显示图片	1.2.156
特殊控件 - 钟表效果实现	1.2.157
特殊控件 - 雷达图（蛛网图、五行图）解决方案	1.2.158
界面设计 - iOS 设备屏幕分辨率	1.2.159
界面设计 - iOS 设计相关图标与图像尺寸	1.2.160
界面设计 - 相关资料收集	1.2.161
相册 - 从系统相册选择照片	1.2.162
系统服务 - 调用系统应用和系统服务	1.2.163
编码规范 - iOS 编码规范(Objective-C & Swift)	1.2.164
编码规范 - 使用 Space Commander 实现 iOS 编码规范自动化	1.2.165
网络 - iOS 访问 HTTPS SSL 和 TLS 双向加密	1.2.166
网络 - iOS7 的多任务处理——后台获取 (Background Fetch)	1.2.167
网络 - 使用 AFNetworking 实现网络请求	1.2.168
网络 - 使用 CTTelephonyNetworkInfo 确定移动网络类型和运营商信息	1.2.169
网络 - 使用 SimplePingHelper 在 iPhone / iPad 上 ping 指定服务	1.2.170

器	1.2.171
网络 - 判断连接状态	1.2.172
设计模式 - iOS 开发常用设计模式简介	1.2.173
设计模式 - 使用命令模式实现撤销删除	1.2.174
设计模式 - 单例模式	1.2.175
语法 - ID类字段生成实现	1.2.176
语法 - If not let - in Swift	1.2.177
语法 - iOS 消息传递机制	1.2.178
语法 - MRC 下全局变量的 Delegate 陷阱	1.2.179
语法 - NSString \ NSData \ NSArray \ NSDictionary 格式转换 ( NSArray / NSDictionary <=> NSData <=> NSString )	1.2.180
语法 - Objective-C Runtime 介绍	1.2.181
语法 - Objective-C 与 Swift 互相调用	1.2.182
语法 - Objective-C 中 BOOL 类成员变量的判断陷阱	1.2.183
语法 - Objective-C 中的 Meta-class 是什么	1.2.184
语法 - Objective-C 基础集合类介绍	1.2.185
语法 - 使用字面量	1.2.186
语法 - 开发常用的宏定义	1.2.187
语法 - 弧度 (radians) 和角度转换 (degree)	1.2.188
语法 - 深入理解 GCD	1.2.189
语法 - 类似 NSError 的引用传值实现	1.2.190
语法 - 通过 registerDefaults 实现单次记录需求	1.2.191
语法 - 随机数生成实现	1.2.192
通知 - iOS7 的多任务处理——远程通知 (Remote Notifications)	
通知 - 本地推送通知 (Local Notification) 的测试与实现	1.2.193
通知 - 远程推送通知 (Remote Notification) 的测试与实现	1.2.194
通讯录 - 访问 iOS 系统通讯录	1.2.196 1.2.195
键盘 - 使用 IQKeyboardManager 完美解决IOS开发键盘遮挡	1.2.197
键盘 - 弹出与收起改变页面高度	1.2.198
音视频 - 使用 POVoiceHUD 实现语音录制	1.2.199
音视频 - 使用 TheAmazingAudioEngine 框架实现 iOS 音频开发中的各种	

音效实现	1.2.200
音视频 - 音频播放、录音、视频播放、拍照、视频录制	1.2.201
Python	1.3
Python - Mac 下安装 Python 模块	1.3.1
Python - Python 工具收集	1.3.2
Python - 使用 Anaconda 进行 Python 科学计算	1.3.3
Python - 使用 virualenv 创建虚拟 Python 环境	1.3.4
Python - 学习资源收集	1.3.5
Python - 用 Python 和 py2app 写独立的 Mac OS X 应用	1.3.6
Python - 编码规范	1.3.7
JavaScript	1.4
JavaScript - 开发工具介绍	1.4.1
JavaScript - JavaScript To Avoid	1.4.2
JavaScript - 使用 PDFObject 实现 PDF 文件在线预览	1.4.3
JavaScript - 正则表达式介绍	1.4.4
专题 - JavaScript 学习资源收集	1.4.5

## 欢迎访问 **iOSNotebook**

这是一本用于记录 iOS 开发过程中遇到的一些问题和解决方案的电子书。

该电子书通过 **GitBook** 工具制作并发布。

该电子书中的大部分解决方案均来自于网络，在编辑过程中尽最大努力保留了所有解决方案的参考链接。

若您发现该电子书中的某些记录侵犯了您的个人权益，可以通过以下方式联系到我，我会立即更新删除。

若您对该电子书中记录的某些解决方案有任何的疑问和建议，也可以通过以下联系方式与我交流。

谢谢！

**QQ : 66745628**

邮箱 : **66745628@qq.com**

主页 : **viktyz.github.io/iosnotebook**

**GitHub : viktyz/iosnotebook**

---

Tips : 您可以通过访问 <https://viktyz.github.io/iosnotebook> 进行文章内容快速查询

---

特别推荐

序号	名称	简介
1	Github - iOS 备忘	收录了 Github 上大量的 iOS 第三方开源代码
2	Skyfox - iOS 开发学习路线图	iOS 开发学习路线图参考
3	Github - Awesome-iOS	Awesome 系列之 iOS
4	Github - iOS 资源大全中文版	Awesome 系列之 iOS 中文版

更多链接参考 [专题 - iOS 开发学习资源参考](#)

---

## 文章列表

1. [AppDelegate - iOS APP Launch Options](#)
2. [Auto Layout - 手动添加 Auto Layout 约束\(Masonry\)](#)
3. [Auto Layout - 约束冲突断点调试方法](#)
4. [Block - 介绍与使用](#)
5. [CoreData - 使用 FYHDBManager 管理 CoreData](#)
6. [Git - iOS 开发 gitignore 文件](#)
7. [Git - 如何 clone git 项目到一个非空目录](#)
8. [Git - 常见错误与解决方案](#)
9. [Git - 恢复被误删除分支](#)
10. [JSON - 使用 JSONHelper 进行 JSON 数据解析](#)
11. [Keychain - 使用 PDKeychainBindingsController 实现 Keychain 保存数据封装](#)
12. [KVO - 使用 KVO 更新 UITableViewCell 显示](#)
13. [Mac 系统 - Mac OS 运行shell脚本](#)
14. [Mac 系统 - 使用 Automator 批量重命名](#)
15. [Mac 系统 - 使用 Scp 命令上传下载文件](#)
16. [Mac 系统 - 修改 Mac OS X root 密码](#)
17. [Mac 系统 - 启用定时自动执行任务](#)
18. [Mac 系统 - 通过 Shell 脚本自动记录 Terminal 活动](#)
19. [NSArray - 对自定义对象的数组进行排序](#)
20. [NSData - NSData 数据压缩与解压缩](#)
21. [NSDate - 时间日期相关操作方法总结](#)

- 22. NSMutableSet - 在NSMutableSet中添加自定义对象时怎么保证不重复
- 23. NSObject - 实现自定义对象 isEqual 方法
- 24. NSString - 删除 NSString 中前后空格或回车符
- 25. NSString - 删除 NSString 中特定字符
- 26. NSString - 汉字繁体简体相互转换的实现
- 27. NSString - 汉字转为拼音显示的实现
- 28. NSString - 相似度检测
- 29. NSString - 筛选出 NSString 中特定字符串
- 30. NSTimer - 解决 NSTimer 的循环引用问题导致的内存泄漏
- 31. NSURL - 特殊字符处理
- 32. openURL - iOS App 跳转 App Store 下载和 App Store 评论
- 33. Runtime - 使用 Aspects 实现 Method Swizzling 和 AOP 实践
- 34. UDID - 解决方案介绍与比较
- 35. UIButton - badge 显示的实现(使用 UIBarButtonItem-Badge)
- 36. UIButton - 使用 RNLoadingButton 实现等待按钮
- 37. UIButton - 避免多次重复点击
- 38. UIDevice - 通过 DeviceKit 更方便的获取硬件设备参数
- 39. UIEvent - iOS 事件拦截
- 40. UIImage - 获取 UIImage 大小与压缩 UIImage
- 41. UIImageView - 使用 LBBlurredImage 实现图像模糊效果
- 42. UILabel - UILabel 边缘黑线问题
- 43. UILabel - 悬浮美金单位效果的实现
- 44. UILabel - 显示多格式文本
- 45. UILabel - 显示换行的方法
- 46. UILabel - 计算文本高度
- 47. UILabel - 通过字符串长度计算显示框大小的方法
- 48. UILabel \ UITextField - 设置行间距等富文本属性
- 49. UINavigationBar - 自定义按钮和标题
- 50. UINavigationController - 使用 SlideNavigationController 实现侧滑需求
- 51. UIPageControl - 翻页显示的实现
- 52. UIScrollView - UIScrollView 添加 UIButton 时不响应滑动操作
- 53. UIScrollView - 给 UIScrollView 添加 Autolayout 约束条件
- 54. UISearchBar - 通过 UISearchDisplayViewController 实现全屏搜索显示效果
- 55. UITableView - 使用 EGORefreshTableHeaderView 实现下拉刷新 UITableView
- 56. UITableView - 滑动加载性能优化总结
- 57. UITableView \ UIScrollView - 通过 contentInset 属性设置周边间距

58. UITableViewCell - 动态修改 UITableViewCell 高度
59. UITextField - 使用备忘
60. UITextView - 添加内容后自动滚动到最后一行
61. UIView - iOS 完全复制一个 UIView
62. UIView - 代码实现截图功能
63. UIView - 代码设置 UIView 的阴影、圆角、边框效果
64. UIView \ UIButton - 独占响应事件
65. UIViewController - 旋转问题 willRotateToInterfaceOrientation 方法无法正常调用
66. UIWebView - 使用 UIWebViewToFile 实现 UIWebView 内容转为 Image 或 PDF
67. Xcode - iOS 开发崩溃(Crash)调试
68. Xcode - Xcodeproj 介绍
69. Xcode - 为 Xcode 添加 Snippets
70. Xcode - 使用 BBUncrustifyPlugin-Xcode 插件实现代码格式化
71. Xcode - 使用 chisel 提高 LLDB 调试效率
72. Xcode - 使用 DWURecyclingAlert 进行 UITableView 和 UICollectionView 的绘图性能测试
73. Xcode - 使用 Instruments 的 Allocations 工具检测内存分配情况
74. Xcode - 使用 Instruments 的 Core Animation 工具进行 UIKit 性能调优
75. Xcode - 使用 Jenkins 进行 App 自动化打包
76. Xcode - 使用 LLDB 调试代码
77. Xcode - 在 Xcode 里添加和使用 Debug 宏定义
78. Xcode - 插件开发学习备忘
79. Xcode - 插件的安装与使用
80. Xcode - 系统警告处理( 清除和添加 )
81. Xcode - 调试相关
82. Xcode - 运行时环境变量( Environment Variables )
83. Xcode - 非 App Store 下载官方 Xcode 安装包方法
84. Xib - 通过 Xib 加载自定义 Cell
85. Xib - 通过 Xib 加载自定义 View
86. 专题 - iOS 使用 XMPP 实现即时聊天
87. 专题 - iOS 应用证书、真机调试与发布 Appstore
88. 专题 - iOS 开发学习资源参考
89. 专题 - iOS 开发工具软件集合
90. 专题 - iOS 开发第三方代码库参考

91. 专题 - iOS 开发调试代码集合
92. 专题 - iOS 设备处理器指令集参考
93. 专题 - Mac & iOS 完整开源项目源代码参考
94. 专题 - objc.io Issue 3: 视图 相关文章五篇
95. 专题 - 优秀技术博客参考
96. 专题 - 优秀技术文章参考
97. 专题 - 值得模块化的业务需求
98. 专题 - 创建自己的 iOS 框架
99. 专题 - 可视控件介绍与可定制替代方案推荐
00. 专题 - 后台模式开发指南
01. 专题 - 控制器介绍
02. 专题 - 时间戳的介绍与使用
03. 二维码 - QRCode 生成与识别
04. 其他 - RESTful API 设计与最佳实践介绍
05. 其他 - 书籍推荐
06. 其他 - 有趣的网址收集
07. 内购 - iOS 内购的快速实现
08. 动画 - Core Animation 之 CABaseAnimation
09. 动画 - Core Animation 之 CATransform3D
10. 动画 - Core Animation 之 Key Path
- |11. 动画 - 使用 POVFrameBuilder 快速实现 UIView 的动画移动和布局
12. 动画 - 页面跳转 - 自定义模态跳转动画
13. 地图 - MKMapView 地图开发相关总结
14. 地图 - 为 Xcode 工程添加自定义初始化位置 (载入 gpx 文件)
15. 地图 - 使用 Eviltransform 进行火星坐标转换 (大陆地区地理坐标偏移问题)
16. 字体 - 添加自定义字体
17. 工具 - Markdown 简明教程
18. 工具 - 使用 Cuttermen 插件进行快速切图
19. 工具 - 使用 iBackupbot 查看非越狱真机应用文件
20. 工具 - 使用 objc\_dep 检查项目中的导入依赖(Import Dependancies)
21. 工具 - 使用 VisualJSON 进行 JSON 格式网络接口 API 测试
22. 工具 - 使用 Xcodebuild 进行命令行打包
23. 应用间通信 - App 之间共享数据的几种方法
24. 应用间通信 - URL Schemes
25. 应用间通信 - 实现应用间互相调用与数据传递
26. 应用间通信 - 文档导入导出实现

27. 应用间通信 - 通过 URL 检测是否安装并打开应用
28. 手势 - 判断点击位置是否在某区域 ( View ) 内
29. 手势 - 实现手势操作介绍
30. 授权 - 使用 ClusterPrePermissions 更加友好的提示授权操作
31. 数据加密 - 3DES 加密(包含JAVA服务器端解密)
32. 数据加密 - AES 加密解密
33. 数据加密 - Base64 编码 ( NSData <=> NSString )
34. 数据加密 - DES 加密(包含PHP服务器端解密)
35. 数据加密 - MD5 加密
36. 数据加密 - RSA 加密(包含JAVA服务器端解密)
37. 数据加密 - 使用 NAVAJO 进行密码安全强度检测
38. 数据存储 - 使用 SSZipArchive 实现文件的压缩和解压缩
39. 数据存储 - 序列化对象
40. 数据存储 - 文件(目录)操作笔记
41. 数据库 - 使用 SQLCipher 进行数据库加密存储
42. 文档 - iOS 人机交互指南
43. 文档 - iOS 项目的目录结构
44. 文档 - 苹果审核未通过问题与解决方案参考
45. 时间 - 使用 Benchmarking 精确测量的代码运行时间
46. 时间 - 倒计时器的实现
47. 时间 - 延时执行解决方案
48. 正则表达式 - 在 iOS 开发中使用正则表达式
49. 测试 - iOS 应用测试 Checklist 以及思维导图
50. 测试 - iOS 自动化测试框架介绍
51. 测试 - 使用 DVR 进行网络请求数据的录制与测试
52. 测试 - 异步函数的单元测试
53. 版本兼容 - 7\8 - 无法正确获取 iPad 横竖屏宽高解决方案
54. 特殊控件 - 使用 DBSphereTagCloud 实现标签云效果
55. 特殊控件 - 使用 PSCyclingImageView 实现自动循环轮播图
56. 特殊控件 - 使用 RecordingCircleOverlayView 实现环形记录仪动画
57. 特殊控件 - 使用 SubjectiveCPhotoPanner 实现动作感应显示图片
58. 特殊控件 - 钟表效果实现
59. 特殊控件 - 雷达图（蛛网图、五行图）解决方案
60. 界面设计 - iOS 设备屏幕分辨率
61. 界面设计 - iOS 设计相关图标与图像尺寸
62. 界面设计 - 相关资料收集

- 63. 相册 - 从系统相册选择照片
- 64. 系统服务 - 调用系统应用和系统服务
- 65. 编码规范 - iOS 编码规范(Objective-C & Swift)
- 66. 编码规范 - 使用 Space Commander 实现 iOS 编码规范自动化
- 67. 网络 - iOS 访问 HTTPS SSL 和 TLS 双向加密
- 68. 网络 - iOS7 的多任务处理——后台获取 (Background Fetch)
- 69. 网络 - 使用 AFNetworking 实现网络请求
- 70. 网络 - 使用 CTTelephonyNetworkInfo 确定移动网络类型和运营商信息
- 71. 网络 - 使用 SimplePingHelper 在 iPhone / iPad 上 ping 指定服务器
- 72. 网络 - 判断连接状态
- 73. 设计模式 - iOS 开发常用设计模式简介
- 74. 设计模式 - 使用命令模式实现撤销删除
- 75. 设计模式 - 单例模式
- 76. 语法 - ID类字段生成实现
- 77. 语法 - If not let - in Swift
- 78. 语法 - iOS 消息传递机制
- 79. 语法 - MRC 下全局变量的 Delegate 陷阱
- 80. 语法 - NSString \ NSData \ NSArray \ NSDictionary 格式转换 ( NSArray / NSDictionary <=> NSData <=> NSString )
- 81. 语法 - Objective-C Runtime 介绍
- 82. 语法 - Objective-C 与 Swift 互相调用
- 83. 语法 - Objective-C 中 BOOL 类成员变量的判断陷阱
- 84. 语法 - Objective-C 中的 Meta-class 是什么
- 85. 语法 - Objective-C 基础集合类介绍
- 86. 语法 - 使用字面量
- 87. 语法 - 开发常用的宏定义
- 88. 语法 - 弧度 (radians) 和角度转换 (degree)
- 89. 语法 - 深入理解 GCD
- 90. 语法 - 类似 NSError 的引用传值实现
- 91. 语法 - 通过 registerDefaults 实现单次记录需求
- 92. 语法 - 随机数生成实现
- 93. 通知 - iOS7 的多任务处理——远程通知 (Remote Notifications)
- 94. 通知 - 本地推送通知 (Local Notification) 的测试与实现
- 95. 通知 - 远程推送通知 (Remote Notification) 的测试与实现
- 96. 通讯录 - 访问 iOS 系统通讯录
- 97. 键盘 - 使用 IQKeyboardManager 完美解决IOS开发键盘遮挡

98. 键盘 - 弹出与收起改变页面高度
  99. 音视频 - 使用 PVoiceHUD 实现语音录制
  00. 音视频 - 使用 TheAmazingAudioEngine 框架实现 iOS 音频开发中的各种音效实现
  01. 音视频 - 音频播放、录音、视频播放、拍照、视频录制
- 

## Python 相关

1. Python - Mac 下安装 Python 模块
  2. Python - Python 工具收集
  3. Python - 使用 Anaconda 进行 Python 科学计算
  4. Python - 使用 virualenv 创建虚拟 Python 环境
  5. Python - 学习资源收集
  6. Python - 用 Python 和 py2app 写独立的 Mac OS X 应用
  7. Python - 编码规范
- 

## JavaScript 相关

1. JavaScript - 开发工具介绍
2. JavaScript - JavaScript To Avoid
3. JavaScript - 使用 PDFObject 实现 PDF 文件在线预览
4. JavaScript - 正则表达式介绍
5. 专题 - JavaScript 学习资源收集

## 变更记录

序号	录入时间	录入人	备注
1	2015-03-18	Alfred Jiang	-
2	2015-12-18	Alfred Jiang	-

## 方案名称

AppDelegate - iOS APP Launch Options

## 关键字

AppDelegate \ UIApplication \ iOS程序结构 \ didFinishLaunchingWithOptions \ 应用启动

## 需求场景

1. 需要对应用启动进行类别区分时

## 参考链接

1. NSHipster - UIApplicationDelegate launch Options
2. Apple documentation - UIApplication Class Reference

## 详细内容

- iOS 程序启动时总会调用 **application:didFinishLaunchingWithOptions:**，其中第二个参数 **launchOptions** 为 **NSDictionary** 类型的对象，里面存储有此程序启动的原因。
- **launchOptions** 中的可能键值见 **UIApplication Class Reference** 的 **Launch Options Keys** 节。
- 若用户直接启动， **lauchOptions** 内无数据；

- 若由其他应用程序通过 **openURL:** 启动，则  
**UIApplicationLaunchOptionsURLKey** 对应的对象为启动 URL (**NSURL**) ,  
**UIApplicationLaunchOptionsSourceApplicationKey** 对应启动的源应用程序的 **bundle ID (NSString)** ;
- 若由本地通知启动，则 **UIApplicationLaunchOptionsLocalNotificationKey** 对应的是为启动应用程序的本地通知对象 (**UILocalNotification**) ;
- 若由远程通知启动，则  
**UIApplicationLaunchOptionsRemoteNotificationKey** 对应的是启动应用程序的远程通知信息 **userInfo(NSDictionary)** ;
- 其他key还有：
  - **UIApplicationLaunchOptionsAnnotationKey**
  - **UIApplicationLaunchOptionsLocationKey**
  - **UIApplicationLaunchOptionsNewsstandDownloadsKey**
- 如果要在启动时，做出一些区分，那就需要在下面的代码做处理。

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    NSURL *url = [options objectForKey:UIApplicationLaunchOptionSURLKey];

    if(url){
        //
    }

    NSString *bundleId = [options objectForKey:UIApplicationLaunchOptionsSourceApplicationKey];

    if(bundleId){
        //
    }

    UILocalNotification * localNotify = [options objectForKey:UIApplicationLaunchOptionsLocalNotificationKey];

    if(localNotify){
        //
    }

    NSDictionary * userInfo = [options objectForKey:UIApplicationLaunchOptionsRemoteNotificationKey];

    if(userInfo){
        //
    }
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2015-04-08	Alfred Jiang	-
2	2015-08-18	Alfred Jiang	更新OC示例
3	2015-12-08	Alfred Jiang	添加 Masonry 相关
4	2015-12-23	Alfred Jiang	-
4	2015-03-08	Alfred Jiang	添加 Masonry 备注相关

## 方案名称

Auto Layout - 手动添加 Auto Layout 约束(Masonry)

## 关键字

Auto Layout \ 约束 \ Constraint \ 自动布局 \ 页面布局 \ Masonry

## 需求场景

- 部分页面或控件的代码载入需要实现代码添加约束

## 参考链接

- [Github - Masonry](#)
- [CSDN - Auto Layout 进阶](#)
- [简书 - 追求Masonry](#)
- [Autolayout的第一次亲密接触](#)
- [有趣的Autolayout示例-Masonry实现](#)

## 详细内容

### 1. 系统提供的代码添加约束方法

- 首先设置 View 的 `translatesAutoresizingMaskIntoConstraints` 的属性为 NO ;

2. 然后将子视图 `addSubview` 到父视图中, 否则在添加约束时会产生编译器警告;
3. 添加约束条件.

### Objective-C 示例

```
- (void)addView:(UIView *)aSubView toView:(UIView *)aPView
{
    aSubView.translatesAutoresizingMaskIntoConstraints = NO;
    [aPView addSubview:aSubView];
    [aPView addConstraint:[NSLayoutConstraint constraintWithItem
        :aSubView attribute:NSLayoutAttributeTop relatedBy:NSLayoutRelationEqual
        toItem:aPView attribute:NSLayoutAttributeTop multiplier:1.0 constant:0]];
    [aPView addConstraint:[NSLayoutConstraint constraintWithItem
        :aSubView attribute:NSLayoutAttributeLeft relatedBy:NSLayoutRelationEqual
        toItem:aPView attribute:NSLayoutAttributeLeft multiplier:1.0 constant:0]];
    [aPView addConstraint:[NSLayoutConstraint constraintWithItem
        :aSubView attribute:NSLayoutAttributeRight relatedBy:NSLayoutRelationEqual
        toItem:aPView attribute:NSLayoutAttributeRight multiplier:1.0 constant:0]];
    [aPView addConstraint:[NSLayoutConstraint constraintWithItem
        :aSubView attribute:NSLayoutAttributeBottom relatedBy:NSLayoutRelationEqual
        toItem:aPView attribute:NSLayoutAttributeBottom multiplier:1.0 constant:0]];
    [aPView layoutSubviews];
}
```

### Swift 示例

```
let bidButton = UIButton();

bidButton.setTranslatesAutoresizingMaskIntoConstraints(false)

self.viewBid.addSubview(bidButton);

self.viewBid.addConstraints([
    NSLayoutConstraint(item: bidButton, attribute: NSLayoutAttribute.Top, relatedBy: NSLayoutRelation.Equal, toItem: self.viewBid, attribute: NSLayoutAttribute.Top, multiplier: 1.0, constant: 0),
    NSLayoutConstraint(item: bidButton, attribute: NSLayoutAttribute.Left, relatedBy: NSLayoutRelation.Equal, toItem: self.viewBid, attribute: NSLayoutAttribute.Left, multiplier: 1.0, constant: 0),
    NSLayoutConstraint(item: bidButton, attribute: NSLayoutAttribute.Right, relatedBy: NSLayoutRelation.Equal, toItem: self.viewBid, attribute: NSLayoutAttribute.Right, multiplier: 1.0, constant: 0),
    NSLayoutConstraint(item: bidButton, attribute: NSLayoutAttribute.Bottom, relatedBy: NSLayoutRelation.Equal, toItem: self.viewBid, attribute: NSLayoutAttribute.Bottom, multiplier: 1.0, constant: 0)
])
```

## 2. 使用 **Masonry** 实现代码添加约束方法

```
- (void)addView:(UIView *)aSubview toView:(UIView *)aPView
{
    [aPView addSubview:aSubview];

    UIEdgeInsets edge = UIEdgeInsetsMake(0, 0, 0, 0);
    [view mas_makeConstraints:^(MASConstraintMaker *make) {
        make.edges.equalTo(view.superview).insets(edge);
    }];
}
```

更多示例请参考 [Github - Masonry](#) 以及备注内容

## 效果图

(无)

## 备注

Masonry 相关优秀博文推荐：

- 标哥的技术博客 - Masonry基本用法
- 标哥的技术博客 - 动画更新约束
- 标哥的技术博客 - remake约束
- 标哥的技术博客 - 整体动画更新约束
- 标哥的技术博客 - 复合约束
- 标哥的技术博客 - 比例 (multipliedBy)
- 标哥的技术博客 - tableviewCell布局
- 标哥的技术博客 - scrollview循环布局
- 标哥的技术博客 - 复杂ScrollView布局
- 标哥的技术博客 - scrollview实战场景

## 变更记录

序号	录入时间	录入人	备注
1	2016-01-05	Alfred Jiang	-

## 方案名称

Auto Layout - 约束冲突断点调试方法

## 关键字

Auto Layout \ 调试 \ 断点 \ 约束冲突

## 需求场景

1. 遇到 Auto Layout 约束冲突时

## 参考链接

1. [NSHint - Autolayout Breakpoints](#)

## 详细内容

当遇到以下 Auto Layout 约束冲突提示时

Unable to simultaneously satisfy constraints.  
Probably at least one of the constraints in the following list is one you don't want.  
Try this:

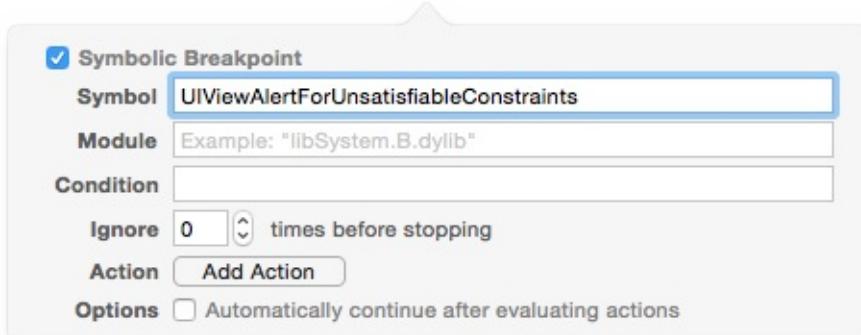
(1) look at each constraint and try to figure out which you don't expect;  
(2) find the code that added the unwanted constraint or constraints and fix it.

(Note: If you're seeing `NSAutoresizingMaskLayoutConstraint`s that you don't understand, refer to the documentation for the `UIView` property `translatesAutoresizingMaskIntoConstraints`)

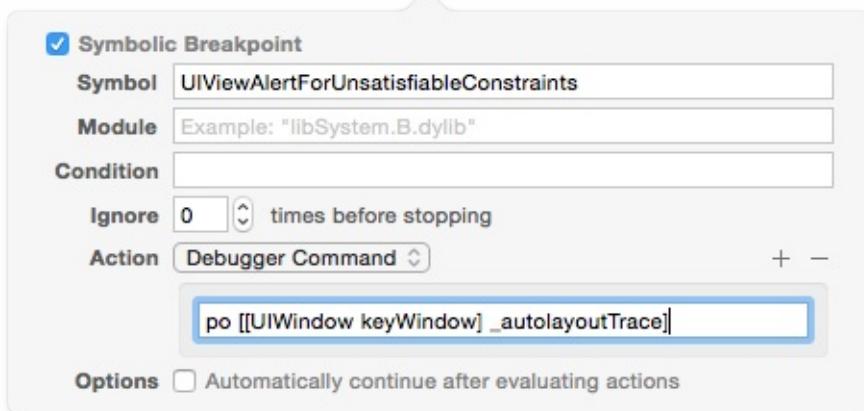
(.....)

Make a symbolic breakpoint at `UIViewAlertForUnsatisfiableConstraints` to catch this in the debugger.  
The methods in the `UIConstraintBasedLayoutDebugging` category on `UIView` listed in `<UIKit/UIKit.h>` may also be helpful.

可以通过添加以下断点定位约束冲突



通过添加以下设置，可以观察到全部的 `UIView` 层级关系以及 **AMBIGUOUS LAYOUT** 提示（**AMBIGUOUS LAYOUT** 即为约束冲突或不完善位置），更好的定位约束冲突



```
UIWindow:0x7f9481c93360
|   •UIView:0x7f9481c9d680
|   |   *UIView:0x7f9481c9d990- AMBIGUOUS LAYOUT for UIView:0x7f9481c9d990.minX{id: 13}, UIView:0x7f9481c9d990.minY{id: 16}
|   |   *_UILayoutGuide:0x7f9481c9e160- AMBIGUOUS LAYOUT for _UILayoutGuide:0x7f9481c9e160.minY{id: 17}
|   |   *_UILayoutGuide:0x7f9481c9ebb0- AMBIGUOUS LAYOUT for _UILayoutGuide:0x7f9481c9ebb0.minY{id: 27}
```

还可以通过 `expr` 命令动态修改页面参数，帮助定位控件位置

```
(lldb) expr ((UIView *)0x7f9ea3d43410).backgroundColor = [UIColor redColor]
(UICachedDeviceRGBColor *) $1 = 0x00007f9ea3d43410
```

## 效果图

(无)

## 备注

- [Xcode - 使用 LLDB 调试代码](#)
- [Xcode - 调试相关](#)



## 变更记录

序号	录入时间	录入人	备注
1	2015-03-02	Alfred Jiang	-
2	2015-12-23	Alfred Jiang	-
3	2016-05-06	Alfred Jiang	更新 dispatch_group_enter() 和 dispatch_group_leave() 成对出现备注说明

## 方案名称

Block - 介绍与使用

## 关键字

Block \ Dispatch \ GCD \ 异步 \ 并行

## 需求场景

1. 单例模式 (`dispatch_once`)
2. 延时执行 (`dispatch_after`)
3. 队列需求 (`dispatch_queue_create`)
4. 孤立队列
5. 迭代执行
6. 组 (`dispatch_group_create`)
7. 事件源
8. 输入输出
9. 基准测试
10. 原子操作
11. 动画
12. 回调
13. 异步

## 参考链接

1. [CocoaChina - Objc的底层并发API](#)
2. [CSDN - 初探swift语言的学习笔记十](#)
3. [《Blocks编程要点》](#)
4. [简书 - 深入研究Block捕获外部变量和\\_\\_block实现原理\(推荐\)](#)
5. [简书 - 深入研究Block用weakSelf、strongSelf、@weakify、@strongify解决循环引用\(推荐\)](#)

## 详细内容

### 1. 如何定义

1. Swift 解决方案 `` swift // 申明 //无参无返回值 typealias funcBlock = () -> () // 或者 () -> Void // 返回值是String typealias funcBlockA = (Int,Int) -> String // 返回值是一个函数指针，入参为String typealias funcBlockB = (Int,Int) -> (String)-> () // 返回值是一个函数指针，入参为String 返回值也是String typealias funcBlockC = (Int,Int) -> (String)->String

```
// 定义 var blockProperty : funcBlockA = {a,b in return ""/**/} // 带初始化方式 var blockPropertyNoReturn : (String) -> () = {param in }
```

```
// 调用 //block作为函数参数 func testBlock(blockfunc:funcBlock!)// 使用!号不需要再解包 { if let exsistblock = blockfunc { blockfunc() //无参无返回 } }
```

### 2. Objective-C 解决方案

```
```objectivec
// 申明
(void) (^loggerBlock)(void);
```

```
// 定义
loggerBlock = ^{
    NSLog(@"Hello world");
};
```

```
// 调用
loggerBlock();
```

## 2. 如何使用

### (1) 异步 API

如果你写一个类，让你类的使用这设置一个将回调传递到的队列会是一个好的选择。你的代码可能像这样

```
- (void)processImage:(UIImage *)image completionHandler:(void(^)(  
BOOL success))handler;  
{  
    dispatch_async(self.isolationQueue, ^(void){  
        // do actual processing here  
        dispatch_async(self.resultQueue, ^(void){  
            handler(YES);  
        });  
    });  
}
```

也可能是这样

```
// 原代码块一
self.indicator.hidden = NO;
[self.indicator startAnimating];
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
    // 原代码块二
    NSURL * url = [NSURL URLWithString:@"http://www.youdao.com"];
;
    NSError * error;
    NSString * data = [NSString stringWithContentsOfURL:url encoding:NSUTF8StringEncoding error:&error];
    if (data != nil) {
        // 原代码块三
        dispatch_async(dispatch_get_main_queue(), ^{
            [self.indicator stopAnimating];
            self.indicator.hidden = YES;
            self.content.text = data;
        });
    } else {
        NSLog(@"error when download:%@", error);
    }
});
```

### Swift 示例

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), { () -> Void in
    dispatch_async(dispatch_get_main_queue(), { () -> Void in
        })
})
```

### (2) 迭代执行

如果你的代码看起来是这样的：

```
for (size_t y = 0; y < height; ++y) {
    for (size_t x = 0; x < width; ++x) {
        // Do something with x and y here
    }
}
```

小小的改动，或许就可以让他运行的更快：

```
dispatch_apply(height, dispatch_get_global_queue(0, 0), ^(size_t
y) {
    for (size_t x = 0; x < width; x += 2) {
        // Do something with x and y here
    }
});
```

### (3) 并行执行的组

```

dispatch_group_t group = dispatch_group_create();

dispatch_queue_t queue = dispatch_get_global_queue(0, 0);
dispatch_group_async(group, queue, ^{
    // Do something that takes a while
    [self doSomeFoo];
    dispatch_group_async(group, dispatch_get_main_queue(), ^{
        self.foo = 42;
    });
});

dispatch_group_async(group, queue, ^{
    // Do something else that takes a while
    [self doSomeBar];
    dispatch_group_async(group, dispatch_get_main_queue(), ^{
        self.bar = 1;
    });
});

// This block will run once everything above is done:
dispatch_group_notify(group, dispatch_get_main_queue(), ^{
    NSLog(@"foo: %ld", (long)self.foo);
    NSLog(@"bar: %ld", (long)self.bar);
});

```

//2015-03-02 13:44:22.961 TestButton[34856:267105] foo: 42  
//2015-03-02 13:44:22.962 TestButton[34856:267105] bar: 1

#### (4) 对现有 API 使用 dispatch\_group\_t

一旦你将组作为你的工具箱中的一部分，你可能会想知道为什么大多数的异步API不把dispatch\_group\_t作为一个可选参数。这没有什么令人绝望的理由，仅仅是因为自己添加这个功能太简单了，但是你还是要小心以确保自己的代码是成对出现的。

举例来说，我们可以给Core Data的-`performBlock:`函数添加上组的功能，那么API会变得像这个样子：

```
- (void)withGroup:(dispatch_group_t)group performBlock:(dispatch_block_t)block
{
    if (group == NULL) {
        [self performBlock:block];
    } else {
        dispatch_group_enter(group);
        [self performBlock:^{
            block();
            dispatch_group_leave(group);
        }];
    }
}
```

这样做允许我们使用`dispatch_group_notify`来运行一段代码，当Core Data上的一堆操作完成以后。很明显，我们可以给NSURLConnection做同样的事情：

```
+ (void)withGroup:(dispatch_group_t)group
    sendAsynchronousRequest:(NSURLRequest *)request
    queue:(NSOperationQueue *)queue
    completionHandler:(void (^)(NSURLResponse *, NSData *, NSError *))handler
{
    if (group == NULL) {
        [self sendAsynchronousRequest:request
                               queue:queue
                           completionHandler:handler];
    } else {
        dispatch_group_enter(group);
        [self sendAsynchronousRequest:request
                               queue:queue
                           completionHandler:^(NSURLResponse *response,
                                              NSData *data, NSError *error){
            handler(response, data, error);
            dispatch_group_leave(group);
        }];
    }
}
```

为了能正常工作，你需要确保：

`dispatch_group_enter()`一定要在`dispatch_group_leave()`之前运行。

`dispatch_group_enter()`和`dispatch_group_leave()`通常是成对出现的（就算有错误产生时）。若通过多个 block 分支回调，确保每个分支的`dispatch_group_leave()`依然与`dispatch_group_enter()`保持成对出现。举例如下：

```
- (void)viewDidLoad {
    [super viewDidLoad];

    NSString * urlString1 = @"...";
    NSString * urlString2 = @"...";

    AFHTTPSessionManager * manager = [AFHTTPSessionManager manager];
}
```

```

dispatch_group_t group = dispatch_group_create();

dispatch_group_enter(group);
[manager GET:urlString1 parameters:nil progress:nil success:^{
    ^(NSURLSessionDataTask * _Nonnull task, id _Nullable responseObject) {
        NSLog(@"finished 1");
        dispatch_group_leave(group);
    } failure:^(NSURLSessionDataTask * _Nullable task, NSError * _Nonnull error) {
        NSLog(@"failed 1 - error = %@", error.localizedDescription);
        dispatch_group_leave(group);
    }];
}

dispatch_group_enter(group);
[manager GET:urlString2 parameters:nil progress:nil success:^{
    ^(NSURLSessionDataTask * _Nonnull task, id _Nullable responseObject) {
        NSLog(@"finished 2");
        dispatch_group_leave(group);
    } failure:^(NSURLSessionDataTask * _Nullable task, NSError * _Nonnull error) {
        NSLog(@"failed 2 - error = %@", error.localizedDescription);
        dispatch_group_leave(group);
    }];
}

dispatch_group_notify(group, dispatch_get_main_queue(), ^{
    NSLog(@"All done");
});
}

```

## (5) 监视进程

如果一些进程正在运行而你想知道他们什么时候存在，GCD能够做到这些。你也可以使用GCD来检测进程什么时候分叉，也就是产生了子进程或者一个信号被传送给进程（比如SIGTERM）。

```
NSRunningApplication *mail = [NSRunningApplication  
    runningApplicationsWithBundleIdentifier:@"com.apple.mail"];  
if (mail == nil) {  
    return;  
}  
pid_t const pid = mail.processIdentifier;  
self.source = dispatch_source_create(DISPATCH_SOURCE_TYPE_PROC,  
pid,  
    DISPATCH_PROC_EXIT, DISPATCH_TARGET_QUEUE_DEFAULT);  
dispatch_source_set_event_handler(self.source, ^{
    NSLog(@"Mail quit.");
});  
dispatch_resume(self.source);
```

当 Mail.app 退出的时候，这个程序会打印出 Mail quit 。

## 效果图

(无)

## 备注

1. **block** 在实现时就会对它引用到的它所在方法中定义的栈变量进行一次只读拷贝，然后在 **block** 块内使用该只读拷贝；
2. 非内联 (**inline**) **block** 不能直接访问 **self**，只能通过将 **self** 当作参数传递到 **block** 中才能使用，并且此时的 **self** 只能通过 **setter** 或 **getter** 方法访问其属性，不能使用句点式方法。但内联 **block** 不受此限制；
3. 使用 **weak-strong dance** 技术来避免循环引用；
4. **block** 内存管理分析：**block** 其实也是一个 **NSObject** 对象，并且在大多数情况下，**block** 是分配在栈上面的，只有当 **block** 被定义为全局变量或 **block** 块中没有引用任何 **automatic** 变量时，**block** 才分配在全局数据段上。**\_block** 变量也是分配在栈上面的。



## 变更记录

序号	录入时间	录入人	备注
1	2015-03-03	Alfred Jiang	-
2	2015-12-22	Alfred Jiang	-

## 方案名称

CoreData - 使用 FYHDBManager 管理 CoreData

## 关键字

CoreData \ 数据库 \ FYHDBManager

## 需求场景

1. 部分轻型小应用的数据库需求

## 参考链接

(无)

## 详细内容

定义

FYHDBHeader.h

```

//  

//  FYHDBHeader.h  

//  GrandJustice  

//  

//  Created by Alfred Jiang on 3/3/15.  

//  Copyright (c) 2015 FYH. All rights reserved.  

//  
  

#ifndef GrandJustice_FYHDBHeader_h
#define GrandJustice_FYHDBHeader_h  
  

#define NAME_OF_SQLITE  @"GrandJustice.sqlite"
#define NAME_OF_MODELD  @"GrandJustice"  
  

//根据实际需要增加实体定义
#define ENTITY_RESULT_ITEM_NAME      @"GJResultItem"
#define ENTITY_GJPLAYER_NAME         @"GJPlayer"  
  

#endif

```

## FYHDBManager.h

```

//  

//  FYHDBManager.h  

//  ALFNote  

//  

//  Created by FYH on 7/22/14.  

//  Copyright (c) 2014 FYH. All rights reserved.  

//  
  

#import <Foundation/Foundation.h>
#import <CoreData/CoreData.h>
#import "FYHDBHeader.h"  
  

typedef void (^FYHDBOperationCompletionBlock) (NSInteger type, NSError *error);  
  

typedef NS_ENUM(NSInteger, ResultType) {
    ResultType_Success = 1,

```

```

    ResultType_Not_Exist = -1000,
    ResultType_Save_Failed,
    ResultType_Delete_Failed,
    ResultType_Clear_Failed,
    ResultType_Fetch_Failed,
};

typedef NS_ENUM(NSInteger, OperationType) {
    OperationType_Save = 0,
    OperationType_Delete,
};

@interface FYHDBManager : NSObject

- (id)init;
- (NSManagedObjectContext *)mainManagedObjectContext;
- (NSManagedObjectModel *)managedObjectModel;
- (NSPersistentStoreCoordinator *)persistentStoreCoordinator;
- (NSURL *)applicationDocumentsDirectory;
- (void)saveManagedObject:(NSManagedObject *)object completion:(FYHDBOperationCompletionBlock)completionBlock;
- (void)deleteDBObject:(id)object completion:(FYHDBOperationCompletionBlock)completionBlock;
- (NSArray *)fetchdataArrayForEntity:(NSString *)entityName
                           byPredicates:(NSPredicate *)predicate
                           sortDescriptors:(NSArray *)sortDescriptors
                         inManagedObjectContext:(NSManagedObjectContext *)context;
@end

```

## FYHDBManager.m

```

// 
//  FYHDBManager.m
//  ALFNote
//
//  Created by FYH on 7/22/14.
//  Copyright (c) 2014 FYH. All rights reserved.

```

```
//  
  
#import "FYHDBManager.h"  
#import <CoreData/CoreData.h>  
  
@interface FYHDBManager()  
{  
    NSManagedObjectContext * __mainManagedObjectContext;  
    NSManagedObjectModel * __managedObjectModel;  
    NSPersistentStoreCoordinator * __persistentStoreCoordinator;  
}  
  
@end  
  
@implementation FYHDBManager  
  
- (id)init  
{  
    self = [super init];  
  
    __mainManagedObjectContext = [self mainManagedObjectContext]  
;  
  
    NSNotificationCenter *nc = [NSNotificationCenter defaultCenter];  
    [nc addObserver:self selector:@selector(mergeChanges:) name:  
NSManagedObjectContextDidSaveNotification object:nil];  
  
    return self;  
}  
  
#pragma mark - CoreData method  
  
- (void)mergeChanges:(NSNotification *)notification {  
    NSManagedObjectContext *savedContext = [notification object]  
;  
  
    // ignore change notifications for the main MOC  
    if (__mainManagedObjectContext == savedContext)  
{
```

```

        return;
    }

    dispatch_sync(dispatch_get_main_queue(), ^{
        [_mainManagedObjectContext mergeChangesFromContextDidSaveNotification:notification];
    });
    //this tells the main thread moc to run on the main thread,
    and merge in the changes there
    //[_mainManagedObjectContext performSelectorOnMainThread:@selector(mergeChangesFromContextDidSaveNotification:) withObject:notification waitUntilDone:YES];
}

- (void)saveContext
{
    NSError *error = nil;
    NSManagedObjectContext *managedObjectContext = self.mainManagedObjectContext;
    if (managedObjectContext != nil) {
        if ([managedObjectContext hasChanges] && ![managedObjectContext save:&error]) {
            // Replace this implementation with code to handle the error appropriately.
            // abort() causes the application to generate a crash log and terminate. You should not use this function in a shipping application, although it may be useful during development.
            NSLog(@"%@", error, [error userInfo]);
            abort();
        }
    }
}

#pragma mark - Core Data stack

-(void)saveManagedObject:(NSManagedObject *)object completion:(FYHDBOperationCompletionBlock)completionBlock
{
    NSManagedObjectContext *moc = object.managedObjectContext;
}

```

```

if (moc == __mainManagedObjectContext) {
    [moc performBlockAndWait:^{
        NSError *error;
        [moc save:&error];
        completionBlock(OperationType_Save, error);
    }];
}
else
{
    NSError *error = [NSError errorWithDomain:@"db" code:0 userInfo:@{NSLocalizedDescriptionKey:@"db save context fault"}];
    completionBlock(OperationType_Save, error);
}

-(void)deleteDBObject:(id)object completion:(FYHDBOperationCompletionBlock)completionBlock
{
    NSManagedObjectContext *moc = ((NSManagedObject *)object).managedObjectContext;
    __block NSError *error;

    [moc performBlockAndWait:^{
        [moc deleteObject:object];
        [moc save:&error];
        completionBlock(OperationType_Delete, error);
    }];
}

#pragma mark -

// Returns the managed object context for the application.
// If the context doesn't already exist, it is created and bound
// to the persistent store coordinator for the application.
- (NSManagedObjectContext *)mainManagedObjectContext
{
    if (__mainManagedObjectContext) {
        return __mainManagedObjectContext;
    }
}

```

```
NSManagedObjectContext * __managedObjectContext = nil;

    NSPersistentStoreCoordinator *coordinator = [self persistent
StoreCoordinator];
    if (coordinator != nil)
    {
        __managedObjectContext = [[NSManagedObjectContext alloc]
initWithConcurrencyType:NSMainQueueConcurrencyType];
        [__managedObjectContext setPersistentStoreCoordinator:co
ordinator];
    }
    return __managedObjectContext;
}

// Returns the managed object model for the application.
// If the model doesn't already exist, it is created from the ap
plication's model.
- (NSManagedObjectModel *)managedObjectModel
{
    if (__managedObjectModel != nil) {
        return __managedObjectModel;
    }
    NSURL *modelURL = [[NSBundle mainBundle] URLForResource:NAME
_OF_MODELD withExtension:@"momd"];
    __managedObjectModel = [[NSManagedObjectModel alloc] initWit
hContentsOfURL:modelURL];
    return __managedObjectModel;
}

// Returns the persistent store coordinator for the application.
// If the coordinator doesn't already exist, it is created and t
he application's store added to it.
- (NSPersistentStoreCoordinator *)persistentStoreCoordinator
{
    if (__persistentStoreCoordinator != nil)
    {
        return __persistentStoreCoordinator;
    }
}
```

```

NSURL *storeURL = [[self applicationDocumentsDirectory] URLByAppendingPathComponent:NAME_OF_SQLITE];
NSLog(@"AMDBManager storeUrl %@", storeURL);
NSError *error = nil;

NSDictionary *options = [NSDictionary dictionaryWithObjectsAndKeys:
                        [NSNumber numberWithBool:YES], NSMigratePersistentStoresAutomaticallyOption,
                        [NSNumber numberWithBool:YES], NSInferMappingModelAutomaticallyOption, nil];

__persistentStoreCoordinator = [[NSPersistentStoreCoordinator alloc] initWithManagedObjectModel:[self managedObjectModel]];
if (![__persistentStoreCoordinator addPersistentStoreWithType:NSSQLiteStoreType configuration:nil URL:storeURL options:options error:&error])
{
    /*
        Replace this implementation with code to handle the error
        or appropriately.
    */
}

```

abort() causes the application to generate a crash log and terminate. You should not use this function in a shipping application, although it may be useful during development. If it is not possible to recover from the error, display an alert panel that instructs the user to quit the application by pressing the Home button.

Typical reasons for an error here include:

- \* The persistent store is not accessible;
- \* The schema for the persistent store is incompatible with current managed object model.

Check the error message to determine what the actual problem was.

If the persistent store is not accessible, there is typically something wrong with the file path. Often, a file URL is pointing into the application's resources directory instead of a

```
writeable directory.
```

If you encounter schema incompatibility errors during development, you can reduce their frequency by:

- \* Simply deleting the existing store:

```
[[NSFileManager defaultManager] removeItemAtURL:storeURL error:nil]
```

- \* Performing automatic lightweight migration by passing the following dictionary as the options parameter:

```
[NSDictionary dictionaryWithObjectsAndKeys:[NSNumber numberWithBool:YES], NSMigratePersistentStoresAutomaticallyOption,
[NSNumber numberWithBool:YES], NSInferMappingModelAutomaticallyOption, nil];
```

Lightweight migration will only work for a limited set of schema changes; consult "Core Data Model Versioning and Data Migration Programming Guide" for details.

```
*/  
NSLog(@"%@", @"Unresolved error %@", error, [error userInfo]);  
abort();  
}  
  
return __persistentStoreCoordinator;  
}  
  
#pragma mark - Application's Documents directory  
  
// Returns the URL to the application's Documents directory.  
- (NSURL *)applicationDocumentsDirectory  
{  
    return [[[NSFileManager defaultManager] URLsForDirectory:NSDocumentDirectory inDomains:NSUserDomainMask] lastObject];  
}  
  
#pragma mark -  
  
#pragma mark - Internal Method
```

```

- (NSArray *)fetchdataArrayForEntity:(NSString *)entityName
    byPredicates:(NSPredicate *)predicate
    sortDescriptors:(NSArray *)sortDescriptiors
    inManagedObjectContext:(NSManagedObjectContext *)c
ontext
{
    __block NSArray *fetchedObjects = nil;

    if (context == __mainManagedObjectContext)
    {
        [context performBlockAndWait:^{
            NSFetchedResultsController *fetchRequest = [[NSFetchedResultsController alloc]
c] init];
            NSEntityDescription *entity = [NSEntityDescription e
ntityForName:entityName inManagedObjectContext:context];
            [fetchRequest setEntity:entity];
            [fetchRequest setPredicate:predicate];
            [fetchRequest setSortDescriptors:sortDescriptiors];

            NSError *error = nil;
            fetchedObjects = [context executeFetchRequest:fetchR
equest error:&error];
            if (fetchedObjects == nil || error) {
                fetchedObjects = nil;
            }
        }];
    }
    else
    {
        NSLog(@"error: fetching from unknown context");
    }

    return fetchedObjects;
}

@end

```

用法

## DBManager.h

```

//  

//  DBManager.h  

//  YLAlbum  

//  

//  Created by FYH on 7/29/14.  

//  Copyright (c) 2014 FYH. All rights reserved.  

//  
  

#import <Foundation/Foundation.h>  

#import "FYHDBManager.h"  

#import "GJResultItem.h"  

#import "GJPlayer.h"  
  

@interface DBManager : FYHDBManager  
  

+ (DBManager *)sharedManager;  
  

#pragma mark - GJResultItem  
  

- (GJResultItem *)createWithAInfo:(NSString *)aAinfo  

                           BInfo:(NSString *)aBinfo  

                           FInfo:(NSString *)aFinfo  

                           GInfo:(NSString *)aGinfo  

                           Version:(NSString *)aVersion  

                           Type:(NSString *)aType;  
  

- (ResultType)deleteGJResultItemById:(NSString *)aId;  

- (ResultType)updateGJResultItem:(GJResultItem *)aGJResultItem;  

- (GJResultItem *)GJResultItemById:(NSString *)aId;  

- (NSArray *)allGJResultItemsByVersion:(NSString *)aVersion andType:(NSString *)aType;  

- (NSArray *)allGJResultItemsByType:(NSString *)aType;  

- (NSArray *)allGJResultItems;  

- (ResultType)clearallValidGJResultItems;  

- (NSArray *)resultItemByItemA:(NSString *)aAinfo ItemB:(NSString *)aBinfo ItemF:(NSString *)aFinfo ItemG:(NSString *)aGinfo forVersion:(NSString *)aVersion;  

- (BOOL)isExistItemA:(NSString *)aAinfo ItemB:(NSString *)aBinfo

```

```

ItemF:(NSString *)aFinfo ItemG:(NSString *)aGinfo forVersion:(  

NSString *)aVersion;

#pragma mark - GJPlayer

- (GJPlayer *)createGJPlayerWithpRoleInfo:(NSString *)apRoleInfo  

pName:(NSString *)apName pPhone:(NSString *)apPhone pRole:(NST  

ring *)apRole;
- (ResultType)deleteGJPlayerById:(NSString *)aId;
- (GJPlayer *)GJPlayerById:(NSString *)aId;
- (ResultType)updateGJPlayer:(GJPlayer *)aGJPlayer;
- (NSArray *)allGJPlayers;
- (ResultType)clearallValidGJPlayers;

#pragma mark - GJTestPlayer

- (GJPlayer *)createGJTestPlayerWithpRoleInfo:(NSString *)apRole  

Info pName:(NSString *)apName pPhone:(NSString *)apPhone pRole:(  

NSString *)apRole;
- (ResultType)deleteGJTestPlayerById:(NSString *)aId;
- (GJPlayer *)GJTestPlayerById:(NSString *)aId;
- (ResultType)updateGJTestPlayer:(GJPlayer *)aGJPlayer;
- (NSArray *)allGJTestPlayers;
- (ResultType)clearallValidGJTestPlayers;

@end

```

## DBManager.m

```

//  

//  DBManager.m  

//  YLAlbum  

//  

//  Created by FYH on 7/29/14.  

//  Copyright (c) 2014 FYH. All rights reserved.  

//  

#import "DBManager.h"

```

```

@implementation DBManager

+ (NSString *)randomIdFor:(NSString *)aTitle {
    NSString *strId = [aTitle stringByAppendingString:[NSString stringWithFormat:@"%f", [[NSDate date] timeIntervalSince1970]] stringByReplacingOccurrencesOfString:@"." withString:@""];
    return [strId stringByAppendingString:[NSString stringWithFormat:@"%u", arc4random_uniform(10000)]];
}

+(DBManager *)sharedManager
{
    static DBManager *sharedManager;

    static dispatch_once_t onceToken;
    dispatch_once(&onceToken, ^{
        sharedManager = [[DBManager alloc] init];
    });

    return sharedManager;
}

#pragma mark - GJResultItem

- (GJResultItem *)createWithAInfo:(NSString *)aAinfo
                           BInfo:(NSString *)aBinfo
                           FInfo:(NSString *)aFinfo
                           GInfo:(NSString *)aGinfo
                           Version:(NSString *)aVersion
                           Type:(NSString *)aType
{
    NSArray *list = [self resultItemByItemA:aAinfo ItemB:aBinfo
                                         ItemF:aFinfo ItemG:aGinfo forVersion:aVersion];
    if (list && [list count] != 0) {
        return list.firstObject;
    }

    NSManagedObjectContext * __managedObjectContext = [self main
                                                 ManagedObjectContext];
}

```

```

GJResultItem *aGJResultItem = (GJResultItem *)[NSEntityDescription
insertNewObjectForEntityForName:ENTITY_RESULT_ITEM_NAME
inManagedObjectContext:_managedObjectContext];

aGJResultItem.rId = [[self class] randomIdFor:@"ResultItem"];
aGJResultItem.rAInfo = aAinfo;
aGJResultItem.rBInfo = aBinfo;
aGJResultItem.rFInfo = aFinfo;
aGJResultItem.rGInfo = aGinfo;
aGJResultItem.rVersion = aVersion;
aGJResultItem.rType = aType;
aGJResultItem.rCount = @"0";
aGJResultItem.rTime = [[NSString stringWithFormat:@"%@", [[NSDate date] timeIntervalSince1970]] stringByReplacingOccurrencesOfString:@"." withString:@""];
aGJResultItem.rEditing = @"0";
aGJResultItem.rValid = @"1";

ResultType type = [self updateGJResultItem:aGJResultItem];

if (type != ResultType_Success) {

    aGJResultItem = nil;

}

return aGJResultItem;
}

- (ResultType)deleteGJResultItemById:(NSString *)aId
{
    GJResultItem *GJResultItem = [self GJResultItemById:aId];

    if (!GJResultItem) {
        return ResultType_Not_Exist;
    }

    ResultType type = ResultType_Success;

    [self deleteDBObject:GJResultItem completion:^(NSInteger typ

```

```

e, NSError *error) {
    if (error) {
        type = ResultType_Delete_Failed;
        NSLog(@"%@",(long)type);
        NSLog(@"Error to Create New GJResultItem!");
    }
}];

return type;
}

- (ResultType)updateGJResultItem:(GJResultItem *)aGJResultItem
{
    ResultType type = ResultType_Success;

    [self saveManagedObject:aGJResultItem completion:^(NSInteger
type, NSError *error) {
        if (error) {
            type = ResultType_Delete_Failed;
            NSLog(@"%@",(long)type);
            NSLog(@"Error to Delete New GJResultItem!");
        }
    }];
}

return type;
}

- (GJResultItem *)GJResultItemById:(NSString *)aId
{
    NSPredicate * filter = [NSPredicate predicateWithFormat:@"rI
d = %@", aId];
    NSArray *array = [self fetchDataArrayForEntity:ENTITY_RESULT
_ITEM_NAME
                                byPredicates:filter
                                sortDescriptors:nil
                                inManagedObjectContext:[self mainMan
agedObjectContext]];
    return array.firstObject;
}

```

```

- (NSArray *)allGJResultItemsByVersion:(NSString *)aVersion andType:(NSString *)aType
{
    NSPredicate * filter = nil;

    filter = [NSPredicate predicateWithFormat:@"rVersion = %@", aVersion];

    NSSortDescriptor *sortDescriptor = nil;

    if ([aType isEqualToString:@"1"])
    {
        sortDescriptor = [[NSSortDescriptor alloc] initWithKey:@"rCount" ascending:NO];
    }
    else
    {
        sortDescriptor = [[NSSortDescriptor alloc] initWithKey:@"rAInfo" ascending:YES];
    }

    NSArray *sortDescriptors = [[NSArray alloc] initWithObjects:sortDescriptor, nil];
}

NSArray *array = [self fetchDataArrayForEntity:ENTITY_RESULT
                                         _ITEM_NAME
                                         byPredicates:filter
                                         sortDescriptors:sortDescriptors
                                         inManagedObjectContext:[self mainManagedObjectContext]];
return array;
}

- (NSArray *)allGJResultItemsByType:(NSString *)aType
{
    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc] initWithKey:@"rTime" ascending:NO];
    NSArray *sortDescriptors = [[NSArray alloc] initWithObjects:sortDescriptor, nil];
}

```

```

    NSPredicate * filter = [NSPredicate predicateWithFormat:@"rT
ype = %@", aType];
    NSArray *array = [self fetchDataArrayForEntity:ENTITY_RESULT
.ITEM_NAME
                                         byPredicates:filter
                                         sortDescriptors:sortDescripto
rs
                                         inManagedObjectContext:[self mainMan
agedObjectContext]];
    return array;
}

- (NSArray *)allGJResultItems
{
    NSFetchedResultsController *request = [[NSFetchedResultsController alloc] init];

    NSEntityDescription *myEntityQuery = [NSEntityDescription
                                         entityForName:ENTITY_R
ESULT_ITEM_NAME
                                         inManagedObjectContext
:[self mainManagedObjectContext]];

    [request setEntity:myEntityQuery];

    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc]
initWithKey:@"rTime" ascending:NO];
    NSArray *sortDescriptors = [[NSArray alloc] initWithObjects:
sortDescriptor,nil];
    [request setSortDescriptors:sortDescriptors];

    NSError *error =nil;
    NSArray *DeviceArr = [[self mainManagedObjectContext] execut
eFetchRequest:request error:&error];
    return DeviceArr;
}

- (ResultType)clearallValidGJResultItems
{
    NSFetchedResultsController *fetch = [[NSFetchedResultsController alloc] init];

```

```

    [fetch setEntity:[NSEntityDescription entityForName:ENTITY_R
ESULT_ITEM_NAME inManagedObjectContext:[self mainManagedObjectContext]
ntext]]];

    ResultType type = ResultType_Success;

    NSError *error =nil;
    NSArray *dbList = [[self mainManagedObjectContext] executeFetchRequest:fetch error:&error];
    if (error) {
        type = ResultType_Fetch_Failed;
        NSLog(@"Error to Create New GJResultItem!");
    }

    for (id object in dbList) {
        [self deleteDBObject:object completion:^(NSInteger type,
NSError *error) {
            if (error) {
                type = ResultType_Clear_Failed;
                NSLog(@"%@",(long)type);
                NSLog(@"Error to Create New GJResultItem!");
            }
        }];
    }

    return type;
}

- (NSArray *)resultItemByItemA:(NSString *)aAinfo ItemB:(NSString
*)aBinfo ItemF:(NSString *)aFinfo ItemG:(NSString *)aGinfo for
Version:(NSString *)aVersion
{
    NSPredicate *filter = [NSPredicate predicateWithFormat:@"rVe
rsion = %@ && rAInfo = %@ && rBInfo = %@ && rFInfo = %@ && rG
Info = %@", aVersion,aAinfo,aBinfo,aFinfo,aGinfo];

    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc]
initWithKey:@"rTime" ascending:NO];
    NSArray *sortDescriptors = [[NSArray alloc] initWithObjects:

```

```

sortDescriptor, nil];

    NSArray *array = [self fetchdataArrayForEntity:ENTITY_RESULT
ITEM_NAME
                                byPredicates:filter
                                sortDescriptors:sortDescripto
rs
                                inManagedObjectContext:[self mainMan
agedObjectContext]];
}

- (BOOL)isExistItemA:(NSString *)aAinfo ItemB:(NSString *)aBinfo
ItemF:(NSString *)aFinfo ItemG:(NSString *)aGinfo forVersion:(
NSString *)aVersion
{
    NSArray *array = [self resultItemByItemA:aAinfo ItemB:aBinfo
ItemF:aFinfo ItemG:aGinfo forVersion:aVersion];
    return (array && array.count > 0);
}

#pragma mark - GJPlayer

- (GJPlayer *)createGJPlayerWithpRoleInfo:(NSString *)apRoleInfo
pName:(NSString *)apName pPhone:(NSString *)apPhone pRole:(NSST
ring *)apRole
{
    NSManagedObjectContext * __managedObjectContext = [self main
ManagedObjectContext];
    GJPlayer *aGJPlayer = (GJPlayer *)[NSEntityDescription inser
tNewObjectForEntityForName:ENTITY_GJPLAYER_NAME
                                inManagedObjectContext:__managedObjectContext];
    aGJPlayer.pId = [[self class] randmIdFor:@"Player"];
    aGJPlayer.pRoleInfo = apRoleInfo;
    aGJPlayer.pName = apName;
    aGJPlayer.pPhone = apPhone;
}

```

```
aGJPlayer.pRole = apRole;
aGJPlayer.pDirection = @"0";
aGJPlayer.pType = @"Normal";

ResultType type = [self updateGJPlayer:aGJPlayer];

if (type != ResultType_Success) {

    aGJPlayer = nil;
}

return aGJPlayer;
}

- (ResultType)deleteGJPlayerById:(NSString *)aId
{
    GJPlayer *aPlayer = [self GJPlayerById:aId];

    if (!aPlayer) {
        return ResultType_Not_Exist;
    }

    ResultType type = ResultType_Success;

    [self deletedDBObject:aPlayer completion:^(NSInteger type, NSError *error) {
        if (error) {
            type = ResultType_Delete_Failed;
            NSLog(@"%@",(long)type);
            NSLog(@"Error to Create New aPlayer!");
        }
    }];
}

return type;
}

- (GJPlayer *)GJPlayerById:(NSString *)aId
{
    NSPredicate * filter = [NSPredicate predicateWithFormat:@"pI
d = %@", aId];
```

```

 NSArray *array = [self fetchdataArrayForEntity:ENTITY_GJPLAY
ER_NAME
                                byPredicates:filter
                                sortDescriptors:nil
                                inManagedObjectContext:[self mainMan
agedObjectContext]];
    return array.firstObject;
}

- (ResultType)updateGJPlayer:(GJPlayer *)aGJPlayer
{
    ResultType type = ResultType_Success;

    [self saveManagedObject:aGJPlayer completion:^(NSInteger typ
e, NSError *error) {
        if (error) {
            type = ResultType_Save_Failed;
            NSLog(@"Error to Delete New GJPlayer!");
        }
    }];
}

return type;
}

- (NSArray *)allGJPlayers
{
    NSFetchRequest *request = [[NSFetchRequest alloc] init];

    NSEntityDescription *myEntityQuery = [NSEntityDescription
                                         entityForName:ENTITY_G
JPLAYER_NAME
                                         inManagedObjectContext
                                         :[self mainManagedObjectContext]];

    [request setEntity:myEntityQuery];

    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc]
                                         initWithKey:@"pId" ascending:NO];
    NSArray *sortDescriptors = [[NSArray alloc] initWithObjects:
                                sortDescriptor,nil];
}

```

```

[request setSortDescriptors:sortDescriptors];

    NSPredicate * filter = [NSPredicate predicateWithFormat:@"p
Type = %@", @"Normal"];
    [request setPredicate:filter ];

    NSError *error =nil;
    NSArray *GJPlayerArr = [[self mainManagedObjectContext] exec
uteFetchRequest:request error:&error];
    return GJPlayerArr;
}

- (ResultType)clearallValidGJPlayers
{
    NSFetchedResultsController *fetch = [[NSFetchedResultsController alloc] init];

    [fetch setEntity:[NSEntityDescription entityForName:ENTITY_G
JPLAYER_NAME inManagedObjectContext:[self mainManagedObjectContext]
xt]]];

    NSPredicate * filter = [NSPredicate predicateWithFormat:@"pT
ype = %@", @"Normal"];
    [fetch setPredicate:filter ];

    ResultType type = ResultType_Success;

    NSError *error =nil;
    NSArray *dbList = [[self mainManagedObjectContext] executeFe
tchRequest:fetch error:&error];
    if (error) {
        type = ResultType_Fetch_Failed;
        NSLog(@"Error to Create New GJPlayer!");
    }

    for (id object in dbList) {
        [self deleteDBObject:object completion:^(NSInteger type,
NSError *error) {
            if (error) {
                type = ResultType_Clear_Failed;
                NSLog(@"Error to Create New GJPlayer!");
            }
        }];
    }
}

```

```

        }
    }];

}

return type;
}

#pragma mark - GJTestPlayer

- (GJPlayer *)createGJTestPlayerWithpRoleInfo:(NSString *)apRole
Info pName:(NSString *)apName pPhone:(NSString *)apPhone pRole:(
NSString *)apRole
{
    NSManagedObjectContext * __managedObjectContext = [self main
ManagedObjectContext];

    GJPlayer *aGJPlayer = (GJPlayer *)[NSEntityDescription inser
tNewObjectForEntityForName:ENTITY_GJPLAYER_NAME inManagedObjectContext:_
managedObjectContext];

    aGJPlayer.pId = [[self class] randomIdFor:@"Player"];
    aGJPlayer.pRoleInfo = apRoleInfo;
    aGJPlayer.pName = apName;
    aGJPlayer.pPhone = apPhone;
    aGJPlayer.pRole = apRole;
    aGJPlayer.pDirection = @"0";
    aGJPlayer.pType = @"Test";

    ResultType type = [self updateGJPlayer:aGJPlayer];

    if (type != ResultType_Success) {

        aGJPlayer = nil;
    }

    return aGJPlayer;
}

- (ResultType)deleteGJTestPlayerById:(NSString *)aId
{

```

```

        return [self deleteGJPlayerById:aId];
    }

    - (GJPlayer *)GJTestPlayerById:(NSString *)aId
    {
        return [self GJPlayerById:aId];
    }

    - (ResultType)updateGJTestPlayer:(GJPlayer *)aGJPlayer
    {
        return [self updateGJPlayer:aGJPlayer];
    }

    - (NSArray *)allGJTestPlayers
    {
        NSFetchRequest *request = [[NSFetchRequest alloc] init];

        NSEntityDescription *myEntityQuery = [NSEntityDescription
                                              entityForName:ENTITY_G
                                              JPLAYER_NAME inManagedObjectContext:[self mainManagedObjectContext]];
    }

    [request setEntity:myEntityQuery];

    NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc]
                                         initWithKey:@"pId" ascending:NO];
    NSArray *sortDescriptors = [[NSArray alloc] initWithObjects:
                                sortDescriptor, nil];
    [request setSortDescriptors:sortDescriptors];

    NSPredicate * filter = [NSPredicate predicateWithFormat:@"pT
ypte = %@", @"Test"];
    [request setPredicate:filter ];

    NSError *error =nil;
    NSArray *GJPlayerArr = [[self mainManagedObjectContext] exec
                            uteFetchRequest:request error:&error];
    return GJPlayerArr;
}

```

```
- (ResultType)clearallValidGJTestPlayers
{
    NSFetchRequest *fetch = [[NSFetchRequest alloc] init];

    [fetch setEntity:[NSEntityDescription entityForName:ENTITY_G
JPLAYER_NAME inManagedObjectContext:[self mainManagedObjectContext]];
x]];

    NSPredicate * filter = [NSPredicate predicateWithFormat:@"pT
ype = %@", @"Test"];
    [fetch setPredicate:filter ];

    ResultType type = ResultType_Success;

    NSError *error =nil;
    NSArray *dbList = [[self mainManagedObjectContext] executeFe
tchRequest:fetch error:&error];
    if (error) {
        type = ResultType_Fetch_Failed;
        NSLog(@"Error to Create New GJPlayer!");
    }

    for (id object in dbList) {
        [self deleteDBObject:object completion:^(NSInteger type,
NSError *error) {
            if (error) {
                type = ResultType_Clear_Failed;
                NSLog(@"Error to Create New GJPlayer!");
            }
        }];
    }

    return type;
}

@end
```

效果图

(无)

## 备注

类似推荐

- GitHub - xudeheng/CoreDataEnvir
- GitHub - hackiftekhar/IQDatabaseManager

## 变更记录

序号	录入时间	录入人	备注
1	2016-03-14	Alfred Jiang	-

## 方案名称

Git - iOS 开发 gitignore 文件

## 关键字

Git \ iOS \ gitignore

## 需求场景

1. 忽略必须要提交的文件

## 参考链接

1. [GitHub - gitignore](#)

## 详细内容

### Objective-C.gitignore & Swift.gitignore

```
# Xcode
#
# gitignore contributors: remember to update Global/Xcode.gitignore,
# Objective-C.gitignore & Swift.gitignore

## Build generated
build/
DerivedData/

## Various settings
```

```
*.pbxuser
!default.pbxuser
*.mode1v3
!default.mode1v3
*.mode2v3
!default.mode2v3
*.perspectivev3
!default.perspectivev3
xcuserdata/

## Other
*.moved-aside
*.xcuserstate

## Obj-C/Swift specific
*.hmap
*.ipa
*.dSYM.zip
*.dSYM

# CocoaPods
#
# We recommend against adding the Pods directory to your .gitignore. However
# you should judge for yourself, the pros and cons are mentioned
# at:
# https://guides.cocoapods.org/using/using-cocoapods.html#should-i-check-the-pods-directory-into-source-control
#
# Pods/

# Carthage
#
# Add this line if you want to avoid checking in source code from Carthage dependencies.
# Carthage/Checkouts

Carthage/Build

# fastlane
```

```
#  
# It is recommended to not store the screenshots in the git repo  
. Instead, use fastlane to re-generate the  
# screenshots whenever they are needed.  
# For more information about the recommended setup visit:  
# https://github.com/fastlane/fastlane/blob/master/fastlane/docs  
/Gitignore.md  
  
fastlane/report.xml  
fastlane/screenshots
```

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2016-07-21	Alfred Jiang	-

## 方案名称

Git - 如何 clone git 项目到一个非空目录

## 关键字

Git \ clone git 非空目录

## 需求场景

1. 对一个主工程的子工程进行单独的 Git 管理时

## 参考链接

1. [OSChina - 如何 clone git 项目到一个非空目录\(推荐\)](#)

## 详细内容

如果我们往一个非空的目录下 clone git 项目，就会提示错误信息：

```
fatal: destination path '.' already exists and is not an empty directory.
```

解决的办法是：

1. 进入非空目录，假设是 /workdir/proj1
2. 执行以下命令

```
git clone --no-checkout https://git.oschina.net/NextApp/platform.git tmp  
mv tmp/.git .    #将 tmp 目录下的 .git 目录移到当前目录  
rmdir tmp  
git reset --hard HEAD
```

然后就可以进行各种正常操作了。

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2016-07-01	Alfred Jiang	-

## 方案名称

Git - 常见错误与解决方案

## 关键字

Git \ 常见错误 \ 解决方案

## 需求场景

1. 使用 Git 时遇到的常见错误解决方案

## 参考链接

(见详细内容)

## 详细内容

--

1. 参考链接

```
error: The following untracked working tree files would be
overwritten by checkout:
```

```
$ git clean -d -fx "" $ git checkout -f another-branch
```

--

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2016-05-26	Alfred Jiang	-

## 方案名称

Git - 恢复被误删除分支

## 关键字

Git \ 恢复删除分支 \ 强制删除 \ 分支恢复 \ 误删除

## 需求场景

- 1. 用于恢复被误删除的 Git 分支

## 参考链接

- 1. [Atlassian - How to restore a deleted branch\(推荐\)](#)

## 详细内容

- 通过以下命令查看操作日志，获取需要恢复的递交 <sha>

```
$ git log -g
```

- 通过以下命令恢复分支 <branch>

```
$ git checkout -b <branch> <sha>
```

如需要恢复 <sha>(35787c7e277f13277aa493dbe053a70fbf47ebd3) 至  
<branch>(recover\_branch) ,执行命令如下

```
$ git log -g  
commit 35787c7e277f13277aa493dbe053a70fbf47ebd3  
...  
  
$ git checkout -b recover_branch 35787c7e277f13277aa493dbe053a70  
fbf47ebd3
```

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2015-04-09	Alfred Jiang	-
2	2015-12-23	Alfred Jiang	-

## 方案名称

JSON - 使用 JSONHelper 进行 JSON 数据解析

## 关键字

JSON \ JSONHelper \ 解析 \ 网络返回数据

## 需求场景

1. 解析由网络返回的 JSON 格式数据

## 参考链接

1. [GitHub - JSONHelper](#)

## 详细内容

### 1. 将 **JSONHelper.swift** 文件加入工程

### 2. 自定义的解析数据模型需要实现 **Deserializable** 协议

1. 示例一 ````swift // 定义 internal struct Movie: Deserializable { var name: String?  
// You can also use let instead of var if you want. var releaseDate: NSDate?  
  
init(data: [String: AnyObject]) { name <-- data["name"] releaseDate <--  
(data["release\_date"], "yyyy-MM-dd") // Refer to the next section for more  
info. } }

```
// 使用 AFHTTPRequestOperationManager().GET( "http://yoursite.com/movies/"
parameters: nil, success: { operation, data in var movies: [Movie]? movies <--
data["movies"] }
```

```
if let movies = movies {
    // Response contained a movies array, and we deserialized it.
    Do what you want here.
} else {
    // Server gave us a response but there was no "movies" key in
    it, so the movies variable
    // is equal to nil. Do some error handling here.
}
```

```
}, failure: { operation, error in // Handle error. })
```

## 2. 示例二

```
```swift
// 
// User.swift
// REX
//
// Created by Alfred Jiang on 3/19/15.
// Copyright (c) 2015 REX. All rights reserved.
//


import UIKit


var _currentUser: User?
var currentUserKey = "currentUser"
var userDidLoginNotification = "userDidLoginNotification"
var userDidLogoutNotification = "userDidLogoutNotification"


internal struct SideStandardProfileRequest: Deserializable {
    var url: String = ""

    init(data: [String: AnyObject]) {
        url <-- data["url"]
    }
}
```

```
}

internal struct Country: Deserializable {
    var code: String = ""

    init(data: [String: AnyObject]) {
        code <-- data["code"]
    }
}

internal struct Location: Deserializable {
    var country: Country?
    var name: String = ""

    init(data: [String: AnyObject]) {
        name <-- data["name"]
        country <-- data["country"]
    }
}

class User : NSObject , Deserializable{
    var id: String!
    var firstName: String?
    var lastName: String?
    var headline: String?
    var profileImageUrl: String?
    var email: String?
    var industry: String?
    var countryCode : String?
    var locationAddress : String?
    var siteStandardProfileRequest: String?

    var data: NSDictionary?

    required init(data: [String: AnyObject])
    {
        self.data = data
        id <-- data["id"]
        firstName <-- data["firstName"]
        lastName <-- data["lastName"]
    }
}
```

```
        headline <-- data["headline"]
        profileImageUrl <-- data["pictureUrl"]
        email <-- data["emailAddress"]
        industry <-- data["industry"]

        var aLocation: Location?
        aLocation <-- data["location"]
        countryCode = aLocation?.country?.code
        locationAddress = aLocation?.name

        var aSiteStandardProfileRequest : SideStandardProfileRequest?
        aSiteStandardProfileRequest <-- data["siteStandardProfileRequest"]

        siteStandardProfileRequest = aSiteStandardProfileRequest
    .url
    }

class var currentUser: User? {
    get {
        if (_currentUser == nil) {
            var data = NSUserDefaults.standardUserDefaults().objectForKey(currentUserKey) as? NSData
            if (data != nil) {
                var dict = NSJSONSerialization.JSONObjectWithData(data!, options: nil, error: nil) as NSDictionary
                _currentUser <-- dict
            }
        }
        return _currentUser
    }
    set(user) {
        _currentUser = user

        if (_currentUser != nil) {
            var data = NSJSONSerialization.dataWithJSONObject(user!.data!, options: nil, error: nil)
            NSUserDefaults.standardUserDefaults().setObject(
                data, forKey: currentUserKey)
        }
    }
}
```

```

        } else {
            NSUserDefaults.standardUserDefaults().setObject(
                nil, forKey: currentUserKey)
        }
        UserDefaults.standardUserDefaults().synchronize()
    }

}

/*
{
    "userID": "10001",
    "userName": "10001",
    "firstName": "4",
    "lastName": "4",
    "displayName": null,
    "email": "aaa@aaa.com",
    "password": "4",
    "address": "4",
    "headline": null,
    "profileImageUrl": null,
    "industry": null,
    "countryCode": null,
    "siteStandardProfileRequest": null,
    "roleId": "4"
}
*/
func toJSONDict() -> NSDictionary
{
    let aDict : NSMutableDictionary = NSMutableDictionary()

    if let id = self.id {
        aDict.setValue(id, forKey: "userName1")
    }

    if let firstName = self.firstName {
        aDict.setValue(firstName, forKey: "firstName")
    }

    if let lastName = self.lastName {
        aDict.setValue(lastName, forKey: "lastName")
    }

    if let headline = self.headline {
        aDict.setValue(headline, forKey: "headline")
    }

    if let profileImageUrl = self.profileImageUrl {
        aDict.setValue(profileImageUrl, forKey: "profileImage")
    }
}

```

```
eUrl")
}

if let email = self.email {
    aDict.setValue(email, forKey: "email")
}

if let industry = self.industry {
    aDict.setValue(industry, forKey: "industry")
}

if let countryCode = self.countryCode {
    aDict.setValue(countryCode, forKey: "countryCode")
}

if let locationAddress = self.locationAddress {
    aDict.setValue(locationAddress, forKey: "address")
}

if let siteStandardProfileRequest = self.siteStandardProfileRequest {
    aDict.setValue(siteStandardProfileRequest, forKey: "siteStandardProfileRequest")
}

return aDict
}
```

### 3. 更多详细用法参考 [GitHub - JSONHelper](#)

## 效果图

(无)

## 备注

- 遵循 `Deserializable` 协议的属性应该是 `option` 类型



## 变更记录

序号	录入时间	录入人	备注
1	2015-04-02	Alfred Jiang	-
2	2015-09-09	Alfred Jiang	-
3	2015-12-23	Alfred Jiang	-

## 方案名称

Keychain - 使用 PDKeychainBindingsController 实现 Keychain 保存数据封装

## 关键字

Keychain \ 秘钥 \ 安全保存用户密码

## 需求场景

1. 需要保存秘钥串类字串，保证在应用被删除时亦不会消失的需求
2. 安全保存用户密码到 keychain 中的需求

## 参考链接

1. [GitHub - PDKeychainBindingsController](#)

## 详细内容

确保保存的信息在应用关闭甚至被删除时依然保存在本机

1. 将以下文件加入工程

```
PDKeychainBindings.h
PDKeychainBindings.m
PDKeychainBindingsController.h
PDKeychainBindingsController.m
```

## 2. 引入头文件

```
#import "PDKeychainBindings.h"
```

```
3. 使用示例 ````objectivec //保存 PDKeychainBindings *bindings= [PDKeychainBindings sharedKeychainBindings]; [bindings setObject: [textField.text stringByReplacingCharactersInRange:range withString:string] forKey:@"passwordString"]; //读取 [passwordField setText:[[PDKeychainBindings sharedKeychainBindings objectForKey:@"passwordString"]]] ````
```

## 效果图

(无)

## 备注

其他可选方案

1. [GitHub - SSKeychain](#)
2. [GitHub - STUtils:STKeychain](#)

## 变更记录

序号	录入时间	录入人	备注
1	2016-05-04	Alfred Jiang	-

## 方案名称

KVO - 使用 KVO 更新 UITableViewCell 显示

## 关键字

KVO \ UITableView \ 观察属性

## 需求场景

- 通过 KVO 实现 UITableViewCell 自更新，避免重复 Reload Cell

## 参考链接

- Stack Overflow - adding KVO to UITableViewCells
- Nachbaur - Back to Basics: Using KVO

## 详细内容

- 在 **ObjectiveModel** 数据模型中定义观察子属性 **KeyPath**, 示例表示要观察 **statusDescription** 子属性

```
//ObjectiveModel.h 中

#define KEY_PATH_FOR_STATUS_DESCRIPTION      @"statusDescription"

...
@property (nonatomic, strong) NSString      *statusDescription;
...
```

2. 定义 *UITableViewCell* 的子类 *HKTimelineCell*, 并实现监听方法 -  
**(void)observeValueForKeyPath:(NSString )keyPath ofObject:(id)object  
change:(NSDictionary )change context:(void )context\***

```
//HKTimelineCell.h 中
...
//需要注册的属性
@property (strong, nonatomic) ObjectiveModel *request;
...

//注册和移除观察接口
- (void)addObserver:(NSObject *)observer forKeyPath:(NSString *)
keyPath options:(NSKeyValueObservingOptions)options context:(void
*)context;
- (void)removeObserver:(NSObject *)observer forKeyPath:(NSString
*)keyPath context:(void *)context;
...

//HKTimelineCell.m 中
@interface NetworkingTestRequestCell()

// 使用 ObservableKeys 保存 keyPath 观察状态，避免重复注册和重复移除（
// 重复移除会导致 crash）
@property (nonatomic, strong) NSMutableSet *observableKeys;

@end

...
// 为 HKTimelineCell 的 request 属性注册观察（观察 request 中 KeyPat
h 为 KEY_PATH_FOR_STATUS_DESCRIPTION 的子属性），此处的 request 为
ObjectiveModel 类型对象注册观察
- (void)addObserver:(NSObject *)observer forKeyPath:(NSString *)
keyPath options:(NSKeyValueObservingOptions)options context:(void
*)context
{
    if ([_observableKeys containsObject:keyPath]) {
        return;
}
```

```

    }

    if (![_ObservableKeys]) {
        _ObservableKeys = [NSMutableSet set];
    }

    [_ObservableKeys addObject:keyPath];

    [self.request addObserver:observer
        forKeyPath:keyPath
        options:options
        context:context];
}

//移除对 request 中 KeyPath 为 KEY_PATH_FOR_STATUS_DESCRIPTION 的子属性的观察
- (void)removeObserver:(NSObject *)observer forKeyPath:(NSString *)keyPath context:(nullable void *)context
{
    if (![_ObservableKeys containsObject:keyPath]) {
        return;
    }

    [self.request removeObserver:observer forKeyPath:keyPath context:context];
    [_ObservableKeys removeObject:keyPath];
}

//观察 HKTimelineCell 的 request 属性中子属性 (KeyPath 为 KEY_PATH_FOR_STATUS_DESCRIPTION) 值的变化并更新 labelStatus 显示，此处的 request 为 ObjectiveModel 类型对象
- (void)observeValueForKeyPath:(NSString *)keyPath ofObject:(id)object change:(NSDictionary *)change context:(void *)context
{
    if ([KEY_PATH_FOR_STATUS_DESCRIPTION isEqualToString:keyPath]) {

        //         NSString *oldObject = [change objectForKey:NSKeyValueChangeOldKey];
        NSString *newObject = [change objectForKey:NSKeyValueCha

```

```
ngeNewKey];  
  
    self.labelStatus.text = newObject;  
}  
}
```

3. 在 `- (void)tableView:(UITableView *)tableView willDisplayCell:(HKTimelineCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath*` 中注册观察

```
//使用到 HKTimelineCell 的 UITableView DataSource  
  
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath  
{  
    // Don't add observers, or the app may crash later when cells are recycled  
}  
  
- (void)tableView:(UITableView *)tableView willDisplayCell:(HKTimelineCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath  
{  
    [cell addObserver:cell forKeyPath:KEY_PATH_FOR_STATUS_DESCRIPTION options:(NSKeyValueObservingOptionOld | NSKeyValueObservingOptionNew) context:(void *)cell];  
}
```

4. 在 `- (void)tableView:(UITableView *)tableView didEndDisplayingCell:(HKTimelineCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath*` 中移除观察

```
- (void)tableView:(UITableView *)tableView didEndDisplayingCell:(HKTimelineCell *)cell forIndexPath:(NSIndexPath *)indexPath
{
    [cell removeObserver:cell forKeyPath:KEY_PATH_FOR_STATUS_DESCRIPTION context:(void *)cell];
}
```

5. 在 `- (void)viewWillDisappear:(BOOL)animated` 中移除全部观察，因为页面离开时 `didEndDisplayingCell` 方法不会被调用

```
- (void)viewWillDisappear:(BOOL)animated
{
    ...
    [self.tableViewMain.visibleCells enumerateObjectsUsingBlock:^{
        (NetworkingTestRequestCell *cell, NSUInteger idx, BOOL * _Nonnull stop) {
            [cell removeObserver:cell forKeyPath:KEY_PATH_FOR_STATUS_DESCRIPTION context:(void *)cell];
        }];
}
```

6. 在 `- (void)viewWillAppear:(BOOL)animated` 中 `reload` 当前显示的 `cells`, 注册被 `viewWillDisappear` 移除的观察，因为 `cell` 使用了 `ObservableKeys` 避免了重复注册，所以不用担心重复注册问题

```
- (void)viewWillAppear:(BOOL)animated
{
    ...
    [self.tableViewMain reloadRowsAtIndexPaths:self.tableViewMain.indexPathsForVisibleRows withRowAnimation:UITableViewRowAnimationNone];
}
```

7. 若要实现正确的 **KVO**, 在对 **KeyPath** 子属性赋值时一定要使用 `setValue:forKey:` 方法

```
[self setValue:[self statusDescription] forKey:KEY_PATH_FOR_STATUS_DESCRIPTION];
```

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2015-03-12	Alfred Jiang	-
2	2015-12-21	Alfred Jiang	-

## 方案名称

Mac 系统 - Mac OS 运行shell脚本

## 关键字

Mac 系统 \ MAC OS \ shell \ 脚本

## 需求场景

1. 通过shell脚本简化流程时

## 参考链接

1. 博客园 - mac终端下运行shell脚本

## 详细内容

mac 终端下运行 shell 脚本

1、写好自己的脚本，

比如 aa.sh

2、打开终端执行，

方法一：输入命令 `./aa.sh` ,

方法二：直接把 aa.sh 拖入到终端里面。

注意事项：

如果报出问题：

*Permission denied*

就是没有权限。

解决办法：

修改该文件 aa.sh 的权限：

使用命令：

*chmod 777 aa.sh*

然后再执行上面第二步的操作，就 OK 。

效果图

(无)

备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2015-03-10	Alfred Jiang	-
2	2015-12-21	Alfred Jiang	-

## 方案名称

Mac 系统 - 使用 Automator 批量重命名

## 关键字

Mac 系统 \ MAC OS \ Automator \ 批量重命名

## 需求场景

1. 需要批量重命名文件时

## 参考链接

1. 强大的机器，Automator工作示例-批量重命名

## 详细内容

1. 打开 Automator
2. 选择“服务”
3. 查询“重命名”
4. 设定重命名规则（应用范围 - 规则）
5. 保存并命名
6. 批量选中文件
7. 右键打开服务
8. 选择命名服务

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2015-04-22	Alfred Jiang	-
2	2015-12-21	Alfred Jiang	-

## 方案名称

Mac 系统 - 使用 Scp 命令上传下载文件

## 关键字

Mac 系统 \ MAC OS \ Scp \ 上传下载文件

## 需求场景

1. 上传本地文件至服务器或者从服务器下载文件

## 参考链接

1. [Linux SSH远程文件/目录传输命令scp](#)

## 详细内容

1. 使用如下的命令上传本地文件至服务器

```
scp -P 22 -r ~/Documents/* viktyz@120.27.34.52:/var/www/html/iOSBook
```

2. 使用如下的命令下载服务器文件至本地

```
scp -P 22 -r viktyz@120.27.34.52:/var/www/html/iOSBook ~/Documents/...
```

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2015-03-02	Alfred Jiang	-
2	2015-12-21	Alfred Jiang	-

## 方案名称

Mac 系统 - 修改 Mac OS X root 密码

## 关键字

Mac 系统 \ MAC OS \ Root \ 权限

## 需求场景

1. 需要系统 root 权限安装工具时

## 参考链接

1. [John Liu Thinks](#)

## 详细内容

方案1：修改 root 密码

```
| sudo passwd
```

回车后输入两次想要给 root 的密码，done.....

表打我。。

想要 root 的 shell 时候直接 su ，给密码，完事， exit 返回正常

```
| Mac:~ john$ sudo passwd Password:
```

Changing password for root.

```
New password: Retype new password: Mac:~ john$ whoami cxlyx Mac:~  
john$ su Password: sh-3.2# whoami root
```

### 方案2：不修改 root 密码

需要 root 权限的 terminal 时候，

```
sudo bash
```

回车，给自己当前用户密码，完事……

唯一缺点就是不能启动root权限的图形界面

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2015-03-23	Alfred Jiang	-

## 方案名称

Mac 系统 - 启用定时自动执行任务

## 关键字

Mac 系统 \ MAC OS \ 定时执行 \ 自动执行

## 需求场景

1. 需要 Mac 系统执行定时自动任务时

## 参考链接

1. Mac OS X launchd plist StartInterval and StartCalendarInterval examples
2. ITEYE - mac 使用 launchctl 定时运行程序
3. CSDN - Mac 上，执行定时脚本：launchctl
4. Apple documentation - launchd.plist

## 详细内容

1. 进入 root 模式，参考 **Mac 系统 - 修改 Mac OS X root 密码**

2. 编写自动执行脚本 **sample.sh**，并赋予执行权限

```
sh-3.2# cd /pathtoyoursh/ sh-3.2# vim sample.sh sh-3.2# chmod 777  
sample.sh
```

输入任意脚本内容，也可以在 sh 中调用 python 或 ruby 脚本

必须确保脚本是正确可执行的

### 3. 编写自动执行配置文件，并赋予执行权限

进入 `LaunchDaemons`，该目录下脚本配置文件在用户未登录情况下依然能够自动执行

```
| sh-3.2# cd /Library/LaunchDaemons/ sh-3.2# vim info.sample.code.plist
```

输入以下内容

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Label</key>
<string>info.sample.code.plist</string>
<key>ProgramArguments</key>
<array>
<string>/pathtoyoursh/sample.sh</string>
</array>
<key>StartCalendarInterval</key>
<dict>
<key>Minute</key>
<integer>0</integer>
<key>Hour</key>
<integer>18</integer>
</dict>
<key>KeepAlive</key>
<false/>
<key>RunAtLoad</key>
<true/>
</dict>
</plist>
```

以上内容表示系统在每天晚上 18:00 自动执行一次 `/pathtoyoursh/sample.sh` 脚本，更多脚本设置内容和格式见参考链接

必须确保所有路径为从根目录开始的绝对路径

可以通过以下命令检测 `plist` 格式是否正确

```
| sh-3.2# plutil -lint info.sample.code.plist
```

设置权限

```
| sh-3.2# chmod 600 info.sample.code.plist
```

### 3. 启用服务

加载服务

```
| sh-3.2# launchctl load info.sample.code.plist
```

卸载服务

```
| sh-3.2# launchctl unload info.sample.code.plist
```

启动服务

```
| sh-3.2# launchctl start info.sample.code.plist
```

关闭服务

```
| sh-3.2# launchctl stop info.sample.code.plist
```

查看全部服务

```
| sh-3.2# launchctl list
```

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2016-09-30	Alfred Jiang	-

## 方案名称

Mac 系统 - 通过 Shell 脚本自动记录 Terminal 活动

## 关键字

Mac 系统 \ Terminal \ 终端

## 需求场景

1. 自动记录 Terminal 活动以便于查找出错的地方

## 参考链接

1. 系统之家 - MAC将Terminal活动记录下来的方法
2. GitHub - fumbles/dotfiles/bin/loggerScript

## 详细内容

### 命令行方案

```
/usr/bin/script ~/Desktop/Terminal.log
```

### 脚本方案

编写以时间为序命名的小脚本来保存每一次对话脚本 `logger.sh`

```
#!/bin/bash
# logger.sh
# Log everything you do in Terminal.

## Formatted date & time string.
FORMATTED_DATE=`/bin/date "+%Y-%m-%d-%H.%M.%S"`

## Archive the previous file
if [ -f ~/Desktop/Logs/Terminal.log ]; then
/bin/cp -f ~/Desktop/Logs/Terminal.log{,.${FORMATTED_DATE}.txt}
fi

## Create Logs Directory
if [ ! -d ~/Desktop/Logs ]; then
mkdir ~/Desktop/Logs
fi

## Begin a new one
/usr/bin/script ~/Desktop/Logs/Terminal.log
```

执行

```
sh logger.sh
```

效果图

(无)

备注

(无)

## n### 变更记录

序号	录入时间	录入人	备注
1	2015-03-25	Alfred Jiang	-
2	2015-12-23	Alfred Jiang	-

## 方案名称

NSArray - 对自定义对象的数组进行排序

## 关键字

NSArray \ 自定义对象数组 \ 数组排序 \ 排序

## 需求场景

1. 需要对自定义对象的数组进行排序时

## 参考链接

1. GitHub - Cocoa Extensions in Swift
2. CSDN - NSString的常用用法

## 详细内容

### 1. Swift 使用示例

1. 示例一

```
self.allList.sortUsingComparator(  
    {  
        (s1:AnyObject!, s2:AnyObject!) -> NSComparisonResu  
lt in  
            var obj1=s1 as CareerCheck  
            var obj2=s2 as CareerCheck  
            if obj1.cItemUpdateTime < obj2.cItemUpdateTime{  
                return NSComparisonResult.OrderedAscending  
            }else{  
                return NSComparisonResult.OrderedDescending  
            }  
    })
```

## 2. 示例二

```
self.auctionList.sortUsingComparator({  
    (s1:AnyObject!, s2:AnyObject!) -> NSComparisonResult in  
  
        var obj1 = s1 as Auction  
        var obj2 = s2 as Auction  
  
        if aKey == "Name"  
        {  
            return obj1.aName.compare(obj2.aName, options: NSStringCompareOptions.NumericSearch | NSStringCompareOptions.CaseInsensitiveSearch, range: nil, locale: nil)  
        }  
        else if aKey == "Time Left"  
        {  
            return obj1.aEndTime.compare(obj2.aEndTime, options: NSStringCompareOptions.NumericSearch, range: nil, locale: nil)  
        }  
        else if aKey == "Lowest Bid"  
        {  
            return obj1.aLowestBid.compare(obj2.aLowestBid, options: NSStringCompareOptions.NumericSearch, range: nil, locale: nil)  
        }  
        else if aKey == "My Bid"  
        {  
            return obj1.aMyBid.compare(obj2.aMyBid, options: NSStringCompareOptions.NumericSearch, range: nil, locale: nil)  
        }  
        else  
        {  
            return obj1.aMyRank.compare(obj2.aMyRank, options: NSStringCompareOptions.NumericSearch, range: nil, locale: nil)  
        }  
    })
```

## 2. Objective-C 使用示例

```
arryA = [arryB sortedArrayUsingComparator: ^(__block id obj1, __block id obj2)
{
    if ([[obj1 index3] integerValue]> [[obj2 index3] integerValue]) {
        return (NSComparisonResult)NSOrderedDescending;
    }
    if ([[obj1 index3] integerValue]< [[obj2 index3] integerValue]) {
        return (NSComparisonResult)NSOrderedAscending;
    }
    return (NSComparisonResult)NSOrderedSame;
}];
```

### 3. 函数使用说明

```
func compare(aString: String, options mask: NSStringCompareOptions = default, range: Range<String.Index>? = default, locale: NSLocale? = default) -> NSComparisonResult
```

这是 *Cocoa* 在 *Swift* 中所添加的扩展之一，用于比较两个字符串

其中 *NSStringCompareOptions* 参数用于设置比较规则

可选项	说明
CaseInsensitiveSearch	不区分大小写比较
LiteralSearch	区分大小写比较
BackwardsSearch	从字符串末尾开始搜索
AnchoredSearch	搜索限制范围的字符串
NumericSearch	按照字符串里的数字为依据，算出顺序。例如 Foo2.txt < Foo7.txt < Foo25.txt
DiacriticInsensitiveSearch	忽略 "-" 符号的比较
WidthInsensitiveSearch	忽略字符串的长度，比较出结果
ForcedOrderingSearch	忽略不区分大小写比较的选项，并强制返回 NSOrderedAscending 或者 NSOrderedDescending
RegularExpressionSearch	只能应用于 rangeOfString:..., stringByReplacingOccurrencesOfString:... 和 replaceOccurrencesOfString:... 方法。使用通用 兼容的比较方法，如果设置此项，可以去掉 NSCaseInsensitiveSearch 和 NSAnchoredSearch

**Range** 参数用于设置比较范围

**NSLocale** 参数用于提供于用户所处地域相关的定制化信息和首选项信息的设置

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2016-10-13	Alfred Jiang	-

## 方案名称

NSData - NSData 数据压缩与解压缩

## 关键字

NSData \ 文件 \ 压缩与解压缩

## 需求场景

1. 需要对 NSData 进行压缩解压缩存取时

## 参考链接

1. [CSDN - IOS 压缩与解压字符串\(文件\)操作\(推荐\)](#)

## 详细内容

NSData+Compressions.h

```

//  

//  NSData+Compresssion.h  

//  JDIPadNetworking  

//  

//  Created by viktyz on 16/10/13.  

//  Copyright © 2016年 Alfred Jiang. All rights reserved.  

//  

#import <Foundation/Foundation.h>  

@interface NSData (Compresssion)  

- (NSData *)compression;  

- (NSData *)deCompresssion;  

@end

```

**NSData+Compresssion.m**

```

//  

//  NSData+Compresssion.m  

//  JDIPadNetworking  

//  

//  Created by viktyz on 16/10/13.  

//  Copyright © 2016年 Alfred Jiang. All rights reserved.  

//  

#import "NSData+Compresssion.h"  

#include <zlib.h>  

@implementation NSData (Compression)  

- (NSData *)compression  

{  

    if ([self length] == 0) return self;  

    unsigned full_length = (unsigned) self.length;  

    unsigned half_length = (unsigned) self.length / 2;

```

```

        NSMutableData *decompressed = [NSMutableData dataWithLength:
full_length + half_length];
    BOOL done = NO;
    int status;

    z_stream strm;
    strm.next_in = (Bytef *)self.bytes;
    strm.avail_in = (uInt) self.length;
    strm.total_out = 0;
    strm.zalloc = Z_NULL;
    strm.zfree = Z_NULL;

    if (inflateInit2(&strm, (15+32)) != Z_OK) return nil;
    while (!done)
    {
        // Make sure we have enough room and reset the lengths.
        if (strm.total_out >= decompressed.length)
            [decompressed increaseLengthBy: half_length];
        strm.next_out = decompressed.mutableBytes + strm.total_out;
        strm.avail_out = (uInt) (decompressed.length - strm.total_out);

        // Inflate another chunk.
        status = inflate (&strm, Z_SYNC_FLUSH);
        if (status == Z_STREAM_END) done = YES;
        else if (status != Z_OK) break;
    }
    if (inflateEnd (&strm) != Z_OK) return nil;

    // Set real length.
    if (done)
    {
        [decompressed setLength: strm.total_out];
        return [NSData dataWithData: decompressed];
    }
    else return nil;
}

- (NSData *)deCompresssion

```

```

{
    if ([self length] == 0) return self;

    z_stream strm;

    strm.zalloc = Z_NULL;
    strm.zfree = Z_NULL;
    strm.opaque = Z_NULL;
    strm.total_out = 0;
    strm.next_in=(Bytef *)self.bytes;
    strm.avail_in = (uInt) self.length;

    // Compressions Levels:
    // Z_NO_COMPRESSION
    // Z_BEST_SPEED
    // Z_BEST_COMPRESSION
    // Z_DEFAULT_COMPRESSION

    if (deflateInit2(&strm, Z_DEFAULT_COMPRESSION, Z_DEFLATED, (
15+16), 8, Z_DEFAULT_STRATEGY) != Z_OK) return nil;

    NSMutableData *compressed = [NSMutableData dataWithLength:16
384]; // 16K chunks for expansion

    do {

        if (strm.total_out >= compressed.length)
            [compressed increaseLengthBy: 16384];

        strm.next_out = compressed.mutableBytes + strm.total_out
;
        strm.avail_out = (uInt) (compressed.length - strm.total_
out);

        deflate(&strm, Z_FINISH);

    } while (strm.avail_out == 0);

    deflateEnd(&strm);
}

```

```
[compressed setLength: strm.total_out];
return [NSData dataWithData:compressed];
}

@end
```

## 效果图

(无)

## 备注

(无)

## 变更记录

序号	录入时间	录入人	备注
1	2015-03-30	Alfred Jiang	-
2	2015-12-23	Alfred Jiang	-

## 方案名称

NSDate - 时间日期相关操作方法总结

## 关键字

NSDate \ 时间操作 \ 日期

## 需求场景

- 需要用到日期显示和时间操作等场景时

## 参考链接

- [GitHub - DateTools](#)

## 详细内容

### 1. 时间戳相关

- 获取当前时间戳 `objective

```
1. 获取当前时间戳 ``objective
```
- (NSTimeInterval)timeBySecond { NSDate\* dat = [NSDate dateWithTimeIntervalSinceNow:0]; return [dat timeIntervalSince1970]; }
- (NSString \*)stringTimeBySecond { return [NSString stringWithFormat:@"%@", [[self class] timeBySecond]]; } ``
- NSDate 与 时间戳 的相互转换

```
// 将 时间戳 转为 NSDate
convenience init(timeIntervalSince1970 secs: NSTimeInterval)
// 将 NSDate 转为 时间戳
var timeIntervalSince1970: NSTimeInterval { get }
```

## 2. 时间工具类推荐

1. NSDate+DateTools.h ```objectivec // Copyright (C) 2014 by Matthew York // //
 Permission is hereby granted, free of charge, to any // person obtaining a
 copy of this software and // associated documentation files (the "Software"),
 to // deal in the Software without restriction, including // without limitation the
 rights to use, copy, modify, merge, // publish, distribute, sublicense, and/or
 sell copies of the // Software, and to permit persons to whom the Software is
 // furnished to do so, subject to the following conditions: // // The above
 copyright notice and this permission notice shall // be included in all copies or
 substantial portions of the Software. // // THE SOFTWARE IS PROVIDED
 "AS IS", WITHOUT WARRANTY OF ANY KIND, // EXPRESS OR IMPLIED,
 INCLUDING BUT NOT LIMITED TO THE WARRANTIES // OF
 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND // /
 NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
 COPYRIGHT HOLDERS // BE LIABLE FOR ANY CLAIM, DAMAGES OR
 OTHER LIABILITY, WHETHER IN AN // ACTION OF CONTRACT, TORT OR
 OTHERWISE, ARISING FROM, OUT OF OR IN // CONNECTION WITH THE
 SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## **ifndef DateToolsLocalizedStrings**

### **define DateToolsLocalizedStrings(key) \**

```
NSLocalizedStringFromTableInBundle(key, @"DateTools", [NSBundle
bundleWithPath:[[NSBundle mainBundle] resourcePath]
stringByAppendingPathComponent:@"DateTools.bundle"]], nil)
```

**endif**

## import

```
static const long long SECONDS_IN_YEAR = 31556900; static const NSInteger
SECONDS_IN_MONTH_28 = 2419200; static const NSInteger
SECONDS_IN_MONTH_29 = 2505600; static const NSInteger
SECONDS_IN_MONTH_30 = 2592000; static const NSInteger
SECONDS_IN_MONTH_31 = 2678400; static const NSInteger
SECONDS_IN_WEEK = 604800; static const NSInteger SECONDS_IN_DAY =
86400; static const NSInteger SECONDS_IN_HOUR = 3600; static const
NSInteger SECONDS_IN_MINUTE = 60; static const NSInteger
MILLISECONDS_IN_DAY = 86400000;
```

@interface NSDate (DateTools)

## pragma mark - Time Ago

- (NSString)*timeAgoSinceDate:(NSDate)date*;
- (NSString)*shortTimeAgoSinceDate:(NSDate)date*;
- (NSString\*)*timeAgoSinceNow*;
- (NSString \*)*shortTimeAgoSinceNow*;
- (NSString )*timeAgoSinceDate:(NSDate )date*;
- (NSString )*timeAgoSinceDate:(NSDate )date numericDates:*  
 (BOOL)*useNumericDates*;
- (NSString )*shortTimeAgoSinceDate:(NSDate )date*;

## pragma mark - Date Components Without Calendar

- (NSInteger)*era*;
- (NSInteger)*year*;
- (NSInteger)*month*;
- (NSInteger)*day*;
- (NSInteger)*hour*;
- (NSInteger)*minute*;

- (NSInteger)second;
- (NSInteger)weekday;
- (NSInteger)weekdayOrdinal;
- (NSInteger)quarter;
- (NSInteger)weekOfMonth;
- (NSInteger)weekOfYear;
- (NSInteger)yearForWeekOfYear;
- (NSInteger)daysInMonth;
- (NSInteger)dayOfYear; -(NSInteger)daysInYear; -(BOOL)isInLeapYear;
- (BOOL)isToday;

## pragma mark - Date Components With Calendar

- (NSInteger)eraWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)yearWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)monthWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)dayWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)hourWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)minuteWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)secondWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)weekdayWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)weekdayOrdinalWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)quarterWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)weekOfMonthWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)weekOfYearWithCalendar:(NSCalendar \*)calendar;
- (NSInteger)yearForWeekOfYearWithCalendar:(NSCalendar \*)calendar;

## pragma mark - Date Editing

## pragma mark Date By Adding

- (NSDate \*)dateByAddingYears:(NSInteger)years;

- (NSDate \*)dateByAddingMonths:(NSInteger)months;
- (NSDate \*)dateByAddingWeeks:(NSInteger)weeks;
- (NSDate \*)dateByAddingDays:(NSInteger)days;
- (NSDate \*)dateByAddingHours:(NSInteger)hours;
- (NSDate \*)dateByAddingMinutes:(NSInteger)minutes;
- (NSDate \*)dateByAddingSeconds:(NSInteger)seconds;

## pragma mark Date By Subtracting

- (NSDate \*)dateBySubtractingYears:(NSInteger)years;
- (NSDate \*)dateBySubtractingMonths:(NSInteger)months;
- (NSDate \*)dateBySubtractingWeeks:(NSInteger)weeks;
- (NSDate \*)dateBySubtractingDays:(NSInteger)days;
- (NSDate \*)dateBySubtractingHours:(NSInteger)hours;
- (NSDate \*)dateBySubtractingMinutes:(NSInteger)minutes;
- (NSDate \*)dateBySubtractingSeconds:(NSInteger)seconds;

## pragma mark - Date Comparison

### pragma mark Time From

-(NSInteger)yearsFrom:(NSDate )date; -(NSInteger)monthsFrom:(NSDate )date; -  
(NSInteger)weeksFrom:(NSDate )date; -(NSInteger)daysFrom:(NSDate )date; -  
(double)hoursFrom:(NSDate )date; -(double)minutesFrom:(NSDate )date; -  
(double)secondsFrom:(NSDate \*)date;

### pragma mark Time From With Calendar

-(NSInteger)yearsFrom:(NSDate )date calendar:(NSCalendar )calendar; -  
(NSInteger)monthsFrom:(NSDate )date calendar:(NSCalendar )calendar; -  
(NSInteger)weeksFrom:(NSDate )date calendar:(NSCalendar )calendar; -  
(NSInteger)daysFrom:(NSDate )date calendar:(NSCalendar )calendar;

## pragma mark Time Until

```
-(NSInteger)yearsUntil; -(NSInteger)monthsUntil; -(NSInteger)weeksUntil; -
(NSInteger)daysUntil; -(double)hoursUntil; -(double)minutesUntil; -
(double)secondsUntil;
```

## pragma mark Time Ago

```
-(NSInteger)yearsAgo; -(NSInteger)monthsAgo; -(NSInteger)weeksAgo; -
(NSInteger)daysAgo; -(double)hoursAgo; -(double)minutesAgo; -
(double)secondsAgo;
```

## pragma mark Earlier Than

```
-(NSInteger)yearsEarlierThan:(NSDate )date; -(NSInteger)monthsEarlierThan:
(NSDate )date; -(NSInteger)weeksEarlierThan:(NSDate )date; -
(NSInteger)daysEarlierThan:(NSDate )date; -(double)hoursEarlierThan:(NSDate
)date; -(double)minutesEarlierThan:(NSDate )date; -(double)secondsEarlierThan:
(NSDate *)date;
```

## pragma mark Later Than

```
-(NSInteger)yearsLaterThan:(NSDate )date; -(NSInteger)monthsLaterThan:
(NSDate )date; -(NSInteger)weeksLaterThan:(NSDate )date; -
(NSInteger)daysLaterThan:(NSDate )date; -(double)hoursLaterThan:(NSDate
)date; -(double)minutesLaterThan:(NSDate )date; -(double)secondsLaterThan:
(NSDate *)date;
```

## pragma mark Comparators

```
-(BOOL)isEarlierThan:(NSDate )date; -(BOOL)isLaterThan:(NSDate )date; -
(BOOL)isEarlierThanOrEqualTo:(NSDate )date; -(BOOL)isLaterThanOrEqualTo:
(NSDate )date;
```

# pragma mark - Formatted Dates

## pragma mark Formatted With Style

```
-(NSString )formattedDateWithStyle:(NSDateFormatterStyle)style; -(NSString
)formattedDateWithStyle:(NSDateFormatterStyle)style timeZone:(NSTimeZone
)timeZone; -(NSString )formattedDateWithStyle:(NSDateFormatterStyle)style
locale:(NSLocale )locale; -(NSString )formattedDateWithStyle:
(NSDateFormatterStyle)style timeZone:(NSTimeZone )timeZone locale:(NSLocale
)locale;
```

## pragma mark Formatted With Format

```
-(NSString )formattedDateWithFormat:(NSString )format; -(NSString
)formattedDateWithFormat:(NSString )format timeZone:(NSTimeZone )timeZone;
-(NSString )formattedDateWithFormat:(NSString )format locale:(NSLocale )locale;
-(NSString )formattedDateWithFormat:(NSString )format timeZone:(NSTimeZone
)timeZone locale:(NSLocale )locale;
```

## pragma mark - Helpers

+ (NSString \*)defaultCalendarIdentifier;

- (void)setDefaultCalendarIdentifier:(NSString \*)identifier; @end ````
- NSDate+DateTools.h ````objectivec // Copyright (C) 2014 by Matthew York // //
 Permission is hereby granted, free of charge, to any // person obtaining a
 copy of this software and // associated documentation files (the "Software"),
 to // deal in the Software without restriction, including // without limitation the
 rights to use, copy, modify, merge, // publish, distribute, sublicense, and/or
 sell copies of the // Software, and to permit persons to whom the Software is
 // furnished to do so, subject to the following conditions: // // The above
 copyright notice and this permission notice shall // be included in all copies or
 substantial portions of the Software. // // THE SOFTWARE IS PROVIDED

"AS IS", WITHOUT WARRANTY OF ANY KIND, // EXPRESS OR IMPLIED,  
 INCLUDING BUT NOT LIMITED TO THE WARRANTIES // OF  
 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND //  
 NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR  
 COPYRIGHT HOLDERS // BE LIABLE FOR ANY CLAIM, DAMAGES OR  
 OTHER LIABILITY, WHETHER IN AN // ACTION OF CONTRACT, TORT OR  
 OTHERWISE, ARISING FROM, OUT OF OR IN // CONNECTION WITH THE  
 SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## import "NSDate+DateTools.h"

```
typedef NS_ENUM(NSUInteger, DTDateComponent){ DTDateComponentEra,
DTDateComponentYear, DTDateComponentMonth, DTDateComponentDay,
DTDateComponentHour, DTDateComponentMinute, DTDateComponentSecond,
DTDateComponentWeekday, DTDateComponentWeekdayOrdinal,
DTDateComponentQuarter, DTDateComponentWeekOfMonth,
DTDateComponentWeekOfYear, DTDateComponentYearForWeekOfYear,
DTDateComponentDayOfYear };
```

```
static const unsigned int allCalendarUnitFlags = NSYearCalendarUnit |
NSQuarterCalendarUnit | NSMonthCalendarUnit | NSWeekOfYearCalendarUnit |
NSWeekOfMonthCalendarUnit | NSDayCalendarUnit | NSHourCalendarUnit |
NSMinuteCalendarUnit | NSSecondCalendarUnit | NSEraCalendarUnit |
NSWeekdayCalendarUnit | NSWeekdayOrdinalCalendarUnit |
NSWeekCalendarUnit;
```

```
static NSString *defaultCalendarIdentifier = nil; static NSCalendar *implicitCalendar
= nil;
```

@implementation NSDate (DateTools)

- (void)load { [self setDefaultCalendarIdentifier:GregorianCalendar]; }

## pragma mark - Time Ago

```
/**
```

- Takes in a date and returns a string with the most convenient unit of time representing
- how far in the past that date is from now. \*
- @param NSDate - Date to be measured from now \*
- @return NSString - Formatted return string \*/
  - (NSString)*timeAgoSinceDate:(NSDate)date*{ return [date timeAgoSinceDate:[NSDate date]]; }

/\*\*

- Takes in a date and returns a shortened string with the most convenient unit of time representing
- how far in the past that date is from now. \*
- @param NSDate - Date to be measured from now \*
- @return NSString - Formatted return string \*/
  - (NSString)*shortTimeAgoSinceDate:(NSDate)date*{ return [date shortTimeAgoSinceDate:[NSDate date]]; }

/\*\*

- Returns a string with the most convenient unit of time representing
- how far in the past that date is from now. \*
- @return NSString - Formatted return string \*/
  - (NSString\*)*timeAgoSinceNow*{ return [self timeAgoSinceDate:[NSDate date]]; }

/\*\*

- Returns a shortened string with the most convenient unit of time representing
- how far in the past that date is from now. \*
- @return NSString - Formatted return string \*/
  - (NSString \*)*shortTimeAgoSinceNow*{ return [self shortTimeAgoSinceDate:[NSDate date]]; }
- (NSString )*timeAgoSinceDate:(NSDate )date*{ return [self timeAgoSinceDate:date numericDates:NO]; }
- (NSString )*timeAgoSinceDate:(NSDate )date numericDates:(BOOL)useNumericDates*{

```

NSCalendar calendar = [NSCalendar currentCalendar]; NSUInteger unitFlags
= NSMinuteCalendarUnit | NSHourCalendarUnit | NSDayCalendarUnit |
NSWeekCalendarUnit | NSMonthCalendarUnit | NSYearCalendarUnit |
NSSecondCalendarUnit; NSDate earliest = [self earlierDate:date]; NSDate
latest = (earliest == self) ? date : self; NSDateComponents components =
[calendar components:unitFlags fromDate:earliest toDate:latest options:0];

//Not Yet Implemented/Optional //The following strings are present in the
translation files but lack logic as of 2014.04.05 //@"Today", @"This week",
@"This month", @"This year" //and @"This morning", @"This afternoon"

if (components.year >= 2) {

    return [self logicLocalizedStringFromFormat:@"%%d %@", years
ago" withValue:components.year];
}

} else if (components.year >= 1) {

    if (useNumericDates) {
        return DateToolsLocalizedStrings(@"1 year ago");
    }

    return DateToolsLocalizedStrings(@"Last year");
}

} else if (components.month >= 2) {

    return [self logicLocalizedStringFromFormat:@"%%d %@", months
ago" withValue:components.month];
}

} else if (components.month >= 1) {

    if (useNumericDates) {
        return DateToolsLocalizedStrings(@"1 month ago");
    }

    return DateToolsLocalizedStrings(@"Last month");
}

```

```
 } else if (components.weekOfMonth >= 2) {  
  
    return [self logicLocalizedStringFromFormat:@"%%d %@",  
           ago" withValue:components.weekOfMonth];  
  
 } else if (components.weekOfMonth >= 1) {  
  
    if (useNumericDates) {  
        return DateToolsLocalizedStrings(@"1 week ago");  
    }  
  
    return DateToolsLocalizedStrings(@"Last week");  
  
 } else if (components.day >= 2) {  
  
    return [self logicLocalizedStringFromFormat:@"%%d %@",  
           a  
go" withValue:components.day];  
  
 } else if (components.day >= 1) {  
  
    if (useNumericDates) {  
        return DateToolsLocalizedStrings(@"1 day ago");  
    }  
  
    return DateToolsLocalizedStrings(@"Yesterday");  
  
 } else if (components.hour >= 2) {  
  
    return [self logicLocalizedStringFromFormat:@"%%d %@",  
           hours  
ago" withValue:components.hour];  
  
 } else if (components.hour >= 1) {  
  
    return DateToolsLocalizedStrings(@"An hour ago");
```

```

} else if (components.minute >= 2) {

    return [self logicLocalizedStringFromFormat:@"%%d %@minute
s ago" withValue:components.minute];
}

} else if (components.minute >= 1) {

    return DateToolsLocalizedStrings(@"A minute ago");
}

} else if (components.second >= 3) {

    return [self logicLocalizedStringFromFormat:@"%%d %@second
s ago" withValue:components.second];
}

} else {

    return DateToolsLocalizedStrings(@"Just now");
}

}

• (NSString )shortTimeAgoSinceDate:(NSDate )date{

    //If shortened formatting is requested, drop the "ago" part of the time ago
    //use abbreviated unit names

    NSCalendar calendar = [NSCalendar currentCalendar]; NSUInteger unitFlags
    = NSMinuteCalendarUnit | NSHourCalendarUnit | NSDayCalendarUnit |
    NSWeekCalendarUnit | NSMonthCalendarUnit | NSYearCalendarUnit |
    NSSecondCalendarUnit; NSDate earliest = [self earlierDate:date]; NSDate
    latest = (earliest == self) ? date : self; NSDateComponents components =
    [calendar components:unitFlags fromDate:earliest toDate:latest options:0];

    if (components.year >= 1) {

```

```
    return [self logicLocalizedStringFromFormat:@"%%d%@y" withValue:components.year];

} else if (components.month >= 1) {

    return [self logicLocalizedStringFromFormat:@"%%d%@M" withValue:components.month];

} else if (components.weekOfMonth >= 1) {

    return [self logicLocalizedStringFromFormat:@"%%d%@w" withValue:components.weekOfMonth];

} else if (components.day >= 1) {

    return [self logicLocalizedStringFromFormat:@"%%d%@d" withValue:components.day];

} else if (components.hour >= 1) {

    return [self logicLocalizedStringFromFormat:@"%%d%@h" withValue:components.hour];

} else if (components.minute >= 1) {

    return [self logicLocalizedStringFromFormat:@"%%d%@m" withValue:components.minute];

} else if (components.second >= 3) {

    return [self logicLocalizedStringFromFormat:@"%%d%@s" withValue:components.second];

} else {
```

```

        return [self logicLocalizedStringFromFormat:@"%@", value:components.second];
        //return DateToolsLocalizedStrings(@"Now"); //string not yet translated 2014.04.05
    }
}

```

- (NSString ) *logicLocalizedStringFromFormat:(NSString )format* *withValue:(NSInteger)value{ NSString \* localeFormat = [NSString stringWithFormat:format, [self getLocaleFormatUnderscoresWithValue:value]];* *return [NSString stringWithFormat:DateToolsLocalizedStrings(localeFormat), value]; }*
- (NSString ) *getLocaleFormatUnderscoresWithValue:(double)value{ NSString localeCode = [[NSLocale preferredLanguages] objectAtIndex:0];*  
*// Russian (ru) if([localeCode isEqualToString:@"ru"]) {*

```

        int XY = (int)floor(value) % 100;
        int Y = (int)floor(value) % 10;

        if(Y == 0 || Y > 4 || (XY > 10 && XY < 15)) {
            return @"";
        }

        if(Y > 1 && Y < 5 && (XY < 10 || XY > 20)) {
            return @"_";
        }

        if(Y == 1 && XY != 11) {
            return @ "__";
        }
    }
}

```

```

}
// Add more languages here, which are have specific translation rules...

```

```
    return @""; }
```

## pragma mark - Date Components Without Calendar

```
/**
```

- Returns the era of the receiver. (0 for BC, 1 for AD for Gregorian) \*
- @return NSInteger \*/
  - (NSInteger)era{ return [self componentForDate:self type:DTDateComponentEra calendar:nil]; }

```
/**
```

- Returns the year of the receiver. \*
- @return NSInteger \*/
  - (NSInteger)year{ return [self componentForDate:self type:DTDateComponentYear calendar:nil]; }

```
/**
```

- Returns the month of the year of the receiver. \*
- @return NSInteger \*/
  - (NSInteger)month{ return [self componentForDate:self type:DTDateComponentMonth calendar:nil]; }

```
/**
```

- Returns the day of the month of the receiver. \*
- @return NSInteger \*/
  - (NSInteger)day{ return [self componentForDate:self type:DTDateComponentDay calendar:nil]; }

```
/**
```

- Returns the hour of the day of the receiver. (0-24) \*
- @return NSInteger \*/
  - (NSInteger)hour{ return [self componentForDate:self type:DTDateComponentHour calendar:nil]; }

```
/**  
 * • Returns the minute of the receiver. (0-59) *  
 * • @return NSInteger */  
 *     ○ (NSInteger)minute{ return [self componentForDate:self  
 * type:DTDateComponentMinute calendar:nil]; }  
  
/**  
 * • Returns the second of the receiver. (0-59) *  
 * • @return NSInteger */  
 *     ○ (NSInteger)second{ return [self componentForDate:self  
 * type:DTDateComponentSecond calendar:nil]; }  
  
/**  
 * • Returns the day of the week of the receiver. *  
 * • @return NSInteger */  
 *     ○ (NSInteger)weekday{ return [self componentForDate:self  
 * type:DTDateComponentWeekday calendar:nil]; }  
  
/**  
 * • Returns the ordinal for the day of the week of the receiver. *  
 * • @return NSInteger */  
 *     ○ (NSInteger)weekdayOrdinal{ return [self componentForDate:self  
 * type:DTDateComponentWeekdayOrdinal calendar:nil]; }  
  
/**  
 * • Returns the quarter of the receiver. *  
 * • @return NSInteger */  
 *     ○ (NSInteger)quarter{ return [self componentForDate:self  
 * type:DTDateComponentQuarter calendar:nil]; }  
  
/**  
 * • Returns the week of the month of the receiver. *  
 * • @return NSInteger */  
 *     ○ (NSInteger)weekOfMonth{ return [self componentForDate:self  
 * type:DTDateComponentWeekOfMonth calendar:nil]; }
```

```

/**


- Returns the week of the year of the receiver. *
- @return NSInteger */
  - (NSInteger)weekOfYear{ return [self componentForDate:self type:DTDateComponentWeekOfYear calendar:nil]; }


*/


- I honestly don't know much about this value... *
- @return NSInteger */
  - (NSInteger)yearForWeekOfYear{ return [self componentForDate:self type:DTDateComponentYearForWeekOfYear calendar:nil]; }


*/


- Returns how many days are in the month of the receiver. *
- @return NSInteger */
  - (NSInteger)daysInMonth{ NSCalendar *calendar = [NSCalendar currentCalendar]; NSRange days = [calendar rangeOfUnit:NSDayCalendarUnit
inUnit:NSMonthCalendarUnit
forDate:self];


return days.length; }

*/


- Returns the day of the year of the receiver. (1-365 or 1-366 for leap year) *
- @return NSInteger */
  - (NSInteger)dayOfYear{ return [self componentForDate:self type:DTDateComponentDayOfYear calendar:nil]; }


*/


- Returns how many days are in the year of the receiver. *
- @return NSInteger */ -(NSInteger)daysInYear{ if (self.isInLeapYear) {
  - return 366;


}



```

```

    }

    return 365; }

/**



- Returns whether the receiver falls in a leap year. *
- @return NSInteger / -(BOOL)isInLeapYear{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents *dateComponents = [calendar components:allCalendarUnitFlags fromDate:self];
if (dateComponents.year%400 == 0){

    return YES;

} else if (dateComponents.year%100 == 0){

    return NO;

} else if (dateComponents.year%4 == 0){

    return YES;

}

return NO; }

• -(BOOL)isToday { NSCalendar cal = [NSCalendar currentCalendar];
NSDateComponents components = [cal components:(NSEraCalendarUnit|NSYearCalendarUnit|NSMonthCalendarUnit|NSDayCalendarUnit) fromDate:[NSDate date]]; NSDate today = [cal dateFromComponents:components]; components = [cal components:(NSEraCalendarUnit|NSYearCalendarUnit|NSMonthCalendarUnit|NSDayCalendarUnit) fromDate:self]; NSDate otherDate = [cal dateFromComponents:components];

return [today isEqualToDate:otherDate]; }

```

# pragma mark - Date Components With Calendar

```
/**
```

- Returns the era of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the era (0 for BC, 1 for AD for Gregorian) \*/
  - (NSInteger)eraWithCalendar:(NSCalendar \*)calendar{ return [self componentForDate:self type:DTDateComponentEra calendar:calendar]; }

```
/**
```

- Returns the year of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the year as an integer \*/
  - (NSInteger)yearWithCalendar:(NSCalendar \*)calendar{ return [self componentForDate:self type:DTDateComponentYear calendar:calendar]; }

```
/**
```

- Returns the month of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the month as an integer \*/
  - (NSInteger)monthWithCalendar:(NSCalendar \*)calendar{ return [self componentForDate:self type:DTDateComponentMonth calendar:calendar]; }

```
/**
```

- Returns the day of the month of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the day of the month as an integer \*/
  - (NSInteger)dayWithCalendar:(NSCalendar \*)calendar{ return [self componentForDate:self type:DTDateComponentDay calendar:calendar]; }

```
/**
```

- Returns the hour of the day of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the hour of the day as an integer \*/
  - (NSInteger)hourWithCalendar:(NSCalendar \*)calendar{ return [self componentForDate:self type:DTDateComponentHour calendar:calendar]; }

/\*\*

- Returns the minute of the hour of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the minute of the hour as an integer \*/
  - (NSInteger)minuteWithCalendar:(NSCalendar \*)calendar{ return [self componentForDate:self type:DTDateComponentMinute calendar:calendar]; }

/\*\*

- Returns the second of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the second as an integer \*/
  - (NSInteger)secondWithCalendar:(NSCalendar \*)calendar{ return [self componentForDate:self type:DTDateComponentSecond calendar:calendar]; }

/\*\*

- Returns the weekday of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the weekday as an integer \*/
  - (NSInteger)weekdayWithCalendar:(NSCalendar \*)calendar{ return [self componentForDate:self type:DTDateComponentWeekday calendar:calendar]; }

/\*\*

- Returns the weekday ordinal of the receiver from a given calendar \*
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - represents the weekday ordinal as an integer \*/
  - (NSInteger)weekdayOrdinalWithCalendar:(NSCalendar \*)calendar{

```

        return [self componentForDate:self
type:DTDateComponentWeekdayOrdinal calendar:calendar]; }

/**

• Returns the quarter of the receiver from a given calendar *
• @param calendar NSCalendar - The calendar to be used in the calculation *
• @return NSInteger - represents the quarter as an integer */
    ◦ (NSInteger)quarterWithCalendar:(NSCalendar *)calendar{ return [self
componentForDate:self type:DTDateComponentQuarter
calendar:calendar]; }

/**

• Returns the week of the month of the receiver from a given calendar *
• @param calendar NSCalendar - The calendar to be used in the calculation *
• @return NSInteger - represents the week of the month as an integer */
    ◦ (NSInteger)weekOfMonthWithCalendar:(NSCalendar *)calendar{ return
[self componentForDate:self type:DTDateComponentWeekOfMonth
calendar:calendar]; }

/**

• Returns the week of the year of the receiver from a given calendar *
• @param calendar NSCalendar - The calendar to be used in the calculation *
• @return NSInteger - represents the week of the year as an integer */
    ◦ (NSInteger)weekOfYearWithCalendar:(NSCalendar *)calendar{ return
[self componentForDate:self type:DTDateComponentWeekOfYear
calendar:calendar]; }

/**

• Returns the year for week of the year (???) of the receiver from a given
calendar *
• @param calendar NSCalendar - The calendar to be used in the calculation *
• @return NSInteger - represents the year for week of the year as an integer */
    ◦ (NSInteger)yearForWeekOfYearWithCalendar:(NSCalendar *)calendar{
return [self componentForDate:self
type:DTDateComponentYearForWeekOfYear calendar:calendar]; }

```

```

/**
 * Returns the day of the year of the receiver from a given calendar *
 * @param calendar NSCalendar - The calendar to be used in the calculation *
 * @return NSInteger - represents the day of the year as an integer */
    ○ (NSInteger)dayOfYearWithCalendar:(NSCalendar *)calendar{ return [self
        componentForDate:self type:DTDateComponentDayOfYear
        calendar:calendar]; }

/**
 * Takes in a date, calendar and desired date component and returns the
 * desired NSInteger
 * representation for that component *
 * @param date NSDate - The date to be be mined for a desired component
 * @param component DTDateComponent - The desired component (i.e. year,
 * day, week, etc)
 * @param calendar NSCalendar - The calendar to be used in the processing
 * (Defaults to Gregorian) *
 * @return NSInteger / -(NSInteger)componentForDate:(NSDate )date type:
 * (DTDateComponent)component calendar:(NSCalendar *)calendar{ if
 * (!calendar) {

    calendar = [[self class] implicitCalendar];

}

unsigned int unitFlags = 0;

if (component == DTDateComponentYearForWeekOfYear) { unitFlags =
    NSYearCalendarUnit | NSQuarterCalendarUnit | NSMonthCalendarUnit |
    NSWeekOfYearCalendarUnit | NSWeekOfMonthCalendarUnit |
    NSDayCalendarUnit | NSHourCalendarUnit | NSMinuteCalendarUnit |
    NSSecondCalendarUnit | NSEraCalendarUnit | NSWeekdayCalendarUnit |
    NSWeekdayOrdinalCalendarUnit | NSWeekCalendarUnit |
    NSYearForWeekOfYearCalendarUnit; } else {

```

```
unitFlags = allCalendarUnitFlags;  
}  
  
NSDateComponents *dateComponents = [calendar components:unitFlags  
fromDate:date];  
  
switch (component) {
```

```
case DTDateComponentEra:  
    return [dateComponents era];  
case DTDateComponentYear:  
    return [dateComponents year];  
case DTDateComponentMonth:  
    return [dateComponents month];  
case DTDateComponentDay:  
    return [dateComponents day];  
case DTDateComponentHour:  
    return [dateComponents hour];  
case DTDateComponentMinute:  
    return [dateComponents minute];  
case DTDateComponentSecond:  
    return [dateComponents second];  
case DTDateComponentWeekday:  
    return [dateComponents weekday];  
case DTDateComponentWeekdayOrdinal:  
    return [dateComponents weekdayOrdinal];  
case DTDateComponentQuarter:  
    return [dateComponents quarter];  
case DTDateComponentWeekOfMonth:  
    return [dateComponents weekOfMonth];  
case DTDateComponentWeekOfYear:  
    return [dateComponents weekOfYear];  
case DTDateComponentYearForWeekOfYear:  
    return [dateComponents yearForWeekOfYear];  
case DTDateComponentDayOfYear:  
    return [calendar ordinalityOfUnit:NSDayCalendarUnit inUnit:NSYearCalendarUnit forDate:date];  
default:  
    break;  
  
}  
  
return 0; }
```

## pragma mark - Date Editing

# pragma mark Date By Adding

```
/**
```

- Returns a date representing the receivers date shifted later by the provided number of years. \*
  - @param years NSInteger - Number of years to add \*
  - @return NSDate - Date modified by the number of desired years \*/
- (NSDate )*dateByAddingYears:(NSInteger)years{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents \*components = [[NSDateComponents alloc] init]; [components setYear:years]; return [calendar dateByAddingComponents:components toDate:self options:0]; }*

```
/**
```

- Returns a date representing the receivers date shifted later by the provided number of months. \*
  - @param years NSInteger - Number of months to add \*
  - @return NSDate - Date modified by the number of desired months \*/
- (NSDate )*dateByAddingMonths:(NSInteger)months{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents \*components = [[NSDateComponents alloc] init]; [components setMonth:months]; return [calendar dateByAddingComponents:components toDate:self options:0]; }*

```
/**
```

- Returns a date representing the receivers date shifted later by the provided number of weeks. \*
  - @param years NSInteger - Number of weeks to add \*
  - @return NSDate - Date modified by the number of desired weeks \*/
- (NSDate )*dateByAddingWeeks:(NSInteger)weeks{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents \*components = [[NSDateComponents alloc] init]; [components setWeekOfMonth:weeks];*

```
return [calendar dateByAddingComponents:components toDate:self
options:0]; }
```

```
/**
```

- Returns a date representing the receivers date shifted later by the provided number of days. \*
- @param years NSInteger - Number of days to add \*
- @return NSDate - Date modified by the number of desired days \*/

```
◦ (NSDate )dateByAddingDays:(NSInteger)days{ NSCalendar calendar =
[[self class] implicitCalendar]; NSDateComponents *components =
[[NSDateComponents alloc] init]; [components setDay:days];
return [calendar dateByAddingComponents:components toDate:self
options:0]; }
```

```
/**
```

- Returns a date representing the receivers date shifted later by the provided number of hours. \*
- @param years NSInteger - Number of hours to add \*
- @return NSDate - Date modified by the number of desired hours \*/

```
◦ (NSDate )dateByAddingHours:(NSInteger)hours{ NSCalendar calendar =
[[self class] implicitCalendar]; NSDateComponents *components =
[[NSDateComponents alloc] init]; [components setHour:hours];
return [calendar dateByAddingComponents:components toDate:self
options:0]; }
```

```
/**
```

- Returns a date representing the receivers date shifted later by the provided number of minutes. \*
- @param years NSInteger - Number of minutes to add \*
- @return NSDate - Date modified by the number of desired minutes \*/

```
◦ (NSDate )dateByAddingMinutes:(NSInteger)minutes{ NSCalendar
calendar = [[self class] implicitCalendar]; NSDateComponents
*components = [[NSDateComponents alloc] init]; [components
setMinute:minutes];
```

```
return [calendar dateByAddingComponents:components toDate:self
options:0]; }
```

```
/**
```

- Returns a date representing the receivers date shifted later by the provided number of seconds. \*
- @param years NSInteger - Number of seconds to add \*
- @return NSDate - Date modified by the number of desired seconds \*/

```
    ◦ (NSDate )dateByAddingSeconds:(NSInteger)seconds{ NSCalendar
        calendar = [[self class] implicitCalendar]; NSDateComponents
        *components = [[NSDateComponents alloc] init]; [components
        setSecond:seconds];
```

```
return [calendar dateByAddingComponents:components toDate:self
options:0]; }
```

## pragma mark Date By Subtracting

```
/**
```

- Returns a date representing the receivers date shifted earlier by the provided number of years. \*
- @param years NSInteger - Number of years to subtract \*
- @return NSDate - Date modified by the number of desired years \*/

```
    ◦ (NSDate )dateBySubtractingYears:(NSInteger)years{ NSCalendar
        calendar = [[self class] implicitCalendar]; NSDateComponents
        components = [[NSDateComponents alloc] init]; [components
        setYear:-1years];
```

```
return [calendar dateByAddingComponents:components toDate:self
options:0]; }
```

```
/**
```

- Returns a date representing the receivers date shifted earlier by the provided number of months. \*
- @param years NSInteger - Number of months to subtract \*
- @return NSDate - Date modified by the number of desired months \*/

- `(NSDate *)dateBySubtractingMonths:(NSInteger)months{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents components = [[NSDateComponents alloc] init]; [components setMonth:-1months];`

```
return [calendar dateByAddingComponents:components toDate:self options:0]; }
```

```
/**
```

- Returns a date representing the receivers date shifted earlier by the provided number of weeks. \*
- @param years NSInteger - Number of weeks to subtract \*
- @return NSDate - Date modified by the number of desired weeks \*/

- `(NSDate *)dateBySubtractingWeeks:(NSInteger)weeks{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents components = [[NSDateComponents alloc] init]; [components setWeekOfMonth:-1weeks];`

```
return [calendar dateByAddingComponents:components toDate:self options:0]; }
```

```
/**
```

- Returns a date representing the receivers date shifted earlier by the provided number of days. \*
- @param years NSInteger - Number of days to subtract \*
- @return NSDate - Date modified by the number of desired days \*/

- `(NSDate *)dateBySubtractingDays:(NSInteger)days{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents components = [[NSDateComponents alloc] init]; [components setDay:-1days];`

```
return [calendar dateByAddingComponents:components toDate:self options:0]; }
```

```
/**
```

- Returns a date representing the receivers date shifted earlier by the provided number of hours. \*
- @param years NSInteger - Number of hours to subtract \*

- @return NSDate - Date modified by the number of desired hours \*/
  - (NSDate )*dateBySubtractingHours:(NSInteger)hours{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents components = [[NSDateComponents alloc] init]; [components setHour:-1hours]; return [calendar dateByAddingComponents:components toDate:self options:0]; }*

/\*\*

- Returns a date representing the receivers date shifted earlier by the provided number of minutes. \*
  - (NSDate )*dateBySubtractingMinutes:(NSInteger)minutes{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents components = [[NSDateComponents alloc] init]; [components setMinute:-1minutes]; return [calendar dateByAddingComponents:components toDate:self options:0]; }*

◦ (NSDate )*dateBySubtractingMinutes:(NSInteger)minutes{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents components = [[NSDateComponents alloc] init]; [components setMinute:-1minutes]; return [calendar dateByAddingComponents:components toDate:self options:0]; }*

return [calendar dateByAddingComponents:components toDate:self options:0]; }

/\*\*

- Returns a date representing the receivers date shifted earlier by the provided number of seconds. \*
  - @param years NSInteger - Number of seconds to subtract \*
  - @return NSDate - Date modified by the number of desired seconds \*/

◦ (NSDate )*dateBySubtractingSeconds:(NSInteger)seconds{ NSCalendar calendar = [[self class] implicitCalendar]; NSDateComponents components = [[NSDateComponents alloc] init]; [components setSecond:-1seconds]; return [calendar dateByAddingComponents:components toDate:self options:0]; }*

return [calendar dateByAddingComponents:components toDate:self options:0]; }

## pragma mark - Date Comparison

# pragma mark Time From

```
/**
```

- Returns an NSInteger representing the amount of time in years between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative.
- Uses the default Gregorian calendar \*
- @param date NSDate - The provided date for comparison \*
- @return NSInteger - The NSInteger representation of the years between receiver and provided date / -(NSInteger)yearsFrom:(NSDate )date{ return [self yearsFrom:date calendar:nil]; }

```
/**
```

- Returns an NSInteger representing the amount of time in months between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative.
- Uses the default Gregorian calendar \*
- @param date NSDate - The provided date for comparison \*
- @return NSInteger - The NSInteger representation of the years between receiver and provided date / -(NSInteger)monthsFrom:(NSDate )date{ return [self monthsFrom:date calendar:nil]; }

```
/**
```

- Returns an NSInteger representing the amount of time in weeks between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative.
- Uses the default Gregorian calendar \*
- @param date NSDate - The provided date for comparison \*
- @return NSInteger - The double representation of the weeks between receiver and provided date / -(NSInteger)weeksFrom:(NSDate )date{ return [self weeksFrom:date calendar:nil]; }

```
/**
```

- Returns an NSInteger representing the amount of time in days between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative.
- Uses the default Gregorian calendar \*
- @param date NSDate - The provided date for comparison \*
- @return NSInteger - The double representation of the days between receiver and provided date / -(NSInteger)daysFrom:(NSDate )date{ return [self daysFrom:date calendar:nil]; }

```
/**
```

- Returns an NSInteger representing the amount of time in hours between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative. \*
- @param date NSDate - The provided date for comparison \*
- @return double - The double representation of the hours between receiver and provided date / -(double)hoursFrom:(NSDate )date{ return ([self timeIntervalSinceDate:date])/SECONDS\_IN\_HOUR; }

```
/**
```

- Returns an NSInteger representing the amount of time in minutes between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative. \*
- @param date NSDate - The provided date for comparison \*
- @return double - The double representation of the minutes between receiver and provided date / -(double)minutesFrom:(NSDate )date{ return ([self timeIntervalSinceDate:date])/SECONDS\_IN\_MINUTE; }

```
/**
```

- Returns an NSInteger representing the amount of time in seconds between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be

- negative. \*
- @param date NSDate - The provided date for comparison \*
- @return double - The double representation of the seconds between receiver and provided date / -(double)secondsFrom:(NSDate )date{ return [self timeIntervalSinceDate:date]; }

## pragma mark Time From With Calendar

```
/**
```

- Returns an NSInteger representing the amount of time in years between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative. \*
- @param date NSDate - The provided date for comparison
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - The double representation of the years between receiver and provided date / -(NSInteger)yearsFrom:(NSDate )date calendar:(NSCalendar \*)calendar{ if (!calendar) {

```
    calendar = [[self class] implicitCalendar];
```

```
}
```

```
NSDate earliest = [self earlierDate:date]; NSDate latest = (earliest == self) ? date : self; NSInteger multiplier = (earliest == self) ? -1 : 1;
NSDateComponents components = [calendar components:NSYearCalendarUnit fromDate:earliest toDate:latest options:0];
return multiplier * components.year; }
```

```
/**
```

- Returns an NSInteger representing the amount of time in months between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative. \*
- @param date NSDate - The provided date for comparison

- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - The double representation of the months between receiver and provided date / -(NSInteger)monthsFrom:(NSDate )date calendar:(NSCalendar \*)calendar{ if (!calendar) {

```
    calendar = [[self class] implicitCalendar];
```

}

```
NSDate earliest = [self earlierDate:date]; NSDate latest = (earliest == self) ?  
date : self; NSInteger multiplier = (earliest == self) ? -1 : 1;  
NSDateComponents components = [calendar  
components:allCalendarUnitFlags fromDate:earliest toDate:latest options:0];  
return multiplier(components.month + 12*components.year); }
```

/\*\*

- Returns an NSInteger representing the amount of time in weeks between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative. \*
- @param date NSDate - The provided date for comparison
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - The double representation of the weeks between receiver and provided date / -(NSInteger)weeksFrom:(NSDate )date calendar:(NSCalendar \*)calendar{ if (!calendar) {

```
    calendar = [[self class] implicitCalendar];
```

}

```
NSDate earliest = [self earlierDate:date]; NSDate latest = (earliest == self) ?  
date : self; NSInteger multiplier = (earliest == self) ? -1 : 1;  
NSDateComponents components = [calendar  
components:NSSecondCalendarUnit fromDate:earliest toDate:latest options:0];  
return multiplier*components.weekOfMonth; }
```

/\*\*

- Returns an NSInteger representing the amount of time in days between the receiver and the provided date.
- If the receiver is earlier than the provided date, the returned value will be negative. \*
- @param date NSDate - The provided date for comparison
- @param calendar NSCalendar - The calendar to be used in the calculation \*
- @return NSInteger - The double representation of the days between receiver and provided date / -(NSInteger)daysFrom:(NSDate )date calendar:(NSCalendar \*)calendar{ if (!calendar) {

```
    calendar = [[self class] implicitCalendar];
```

```
}
```

```
NSDate earliest = [self earlierDate:date]; NSDate latest = (earliest == self) ?  
date : self; NSInteger multiplier = (earliest == self) ? -1 : 1;  
NSDateComponents components = [calendar  
components:NSDayCalendarUnit fromDate:earliest toDate:latest options:0];  
return multiplier*components.day; }
```

## pragma mark Time Until

```
/**
```

- Returns the number of years until the receiver's date. Returns 0 if the receiver is the same or earlier than now. \*
- @return NSInteger representation of years \*/ -(NSInteger)yearsUntil{ return  
[self yearsLaterThan:[NSDate date]]; }

```
/**
```

- Returns the number of months until the receiver's date. Returns 0 if the receiver is the same or earlier than now. \*
- @return NSInteger representation of months \*/ -(NSInteger)monthsUntil{  
return [self monthsLaterThan:[NSDate date]]; }

```
/**
```

- Returns the number of weeks until the receiver's date. Returns 0 if the receiver is the same or earlier than now. \*
- @return NSInteger representation of weeks \*/ -(NSInteger)weeksUntil{ return [self weeksLaterThan:[NSDate date]]; }

/\*\*

- Returns the number of days until the receiver's date. Returns 0 if the receiver is the same or earlier than now. \*
- @return NSInteger representation of days \*/ -(NSInteger)daysUntil{ return [self daysLaterThan:[NSDate date]]; }

/\*\*

- Returns the number of hours until the receiver's date. Returns 0 if the receiver is the same or earlier than now. \*
- @return double representation of hours \*/ -(double)hoursUntil{ return [self hoursLaterThan:[NSDate date]]; }

/\*\*

- Returns the number of minutes until the receiver's date. Returns 0 if the receiver is the same or earlier than now. \*
- @return double representation of minutes \*/ -(double)minutesUntil{ return [self minutesLaterThan:[NSDate date]]; }

/\*\*

- Returns the number of seconds until the receiver's date. Returns 0 if the receiver is the same or earlier than now. \*
- @return double representation of seconds \*/ -(double)secondsUntil{ return [self secondsLaterThan:[NSDate date]]; }

## pragma mark Time Ago

/\*\*

- Returns the number of years the receiver's date is earlier than now. Returns 0 if the receiver is the same or later than now. \*

- @return NSInteger representation of years \*/ -(NSInteger)yearsAgo{ return [self yearsEarlierThan:[NSDate date]]; }
- /\*\*
- Returns the number of months the receiver's date is earlier than now. Returns 0 if the receiver is the same or later than now. \*
  - @return NSInteger representation of months \*/ -(NSInteger)monthsAgo{ return [self monthsEarlierThan:[NSDate date]]; }
- /\*\*
- Returns the number of weeks the receiver's date is earlier than now. Returns 0 if the receiver is the same or later than now. \*
  - @return NSInteger representation of weeks \*/ -(NSInteger)weeksAgo{ return [self weeksEarlierThan:[NSDate date]]; }
- /\*\*
- Returns the number of days the receiver's date is earlier than now. Returns 0 if the receiver is the same or later than now. \*
  - @return NSInteger representation of days \*/ -(NSInteger)daysAgo{ return [self daysEarlierThan:[NSDate date]]; }
- /\*\*
- Returns the number of hours the receiver's date is earlier than now. Returns 0 if the receiver is the same or later than now. \*
  - @return double representation of hours \*/ -(double)hoursAgo{ return [self hoursEarlierThan:[NSDate date]]; }
- /\*\*
- Returns the number of minutes the receiver's date is earlier than now. Returns 0 if the receiver is the same or later than now. \*
  - @return double representation of minutes \*/ -(double)minutesAgo{ return [self minutesEarlierThan:[NSDate date]]; }
- /\*\*
- Returns the number of seconds the receiver's date is earlier than now. Returns 0 if the receiver is the same or later than now. \*

- @return double representation of seconds \*/ -(double)secondsAgo{ return [self secondsEarlierThan:[NSDate date]]; }

## pragma mark Earlier Than

/\*\*

- Returns the number of years the receiver's date is earlier than the provided comparison date.
- Returns 0 if the receiver's date is later than or equal to the provided comparison date. \*
- @param date NSDate - Provided date for comparison \*
- @return NSInteger representing the number of years / -  
 $(NSInteger)yearsEarlierThan:(NSDate )date{ return ABS(MIN([self yearsFrom:date], 0)); }$

/\*\*

- Returns the number of months the receiver's date is earlier than the provided comparison date.
- Returns 0 if the receiver's date is later than or equal to the provided comparison date. \*
- @param date NSDate - Provided date for comparison \*
- @return NSInteger representing the number of months / -  
 $(NSInteger)monthsEarlierThan:(NSDate )date{ return ABS(MIN([self monthsFrom:date], 0)); }$

/\*\*

- Returns the number of weeks the receiver's date is earlier than the provided comparison date.
- Returns 0 if the receiver's date is later than or equal to the provided comparison date. \*
- @param date NSDate - Provided date for comparison \*
- @return NSInteger representing the number of weeks / -  
 $(NSInteger)weeksEarlierThan:(NSDate )date{ return ABS(MIN([self weeksFrom:date], 0)); }$

- ```
/**
  - Returns the number of days the receiver's date is earlier than the provided comparison date.
  - Returns 0 if the receiver's date is later than or equal to the provided comparison date. *
  - @param date NSDate - Provided date for comparison *
  - @return NSInteger representing the number of days / -  
(NSInteger)daysEarlierThan:(NSDate )date{ return ABS(MIN([self  
daysFrom:date], 0)); }/**
  - Returns the number of hours the receiver's date is earlier than the provided comparison date.
  - Returns 0 if the receiver's date is later than or equal to the provided comparison date. *
  - @param date NSDate - Provided date for comparison *
  - @return double representing the number of hours / -  
(double)hoursEarlierThan:(NSDate )date{ return ABS(MIN([self  
hoursFrom:date], 0)); }/**
  - Returns the number of minutes the receiver's date is earlier than the provided comparison date.
  - Returns 0 if the receiver's date is later than or equal to the provided comparison date. *
  - @param date NSDate - Provided date for comparison *
  - @return double representing the number of minutes / -  
(double)minutesEarlierThan:(NSDate )date{ return ABS(MIN([self  
minutesFrom:date], 0)); }/**
  - Returns the number of seconds the receiver's date is earlier than the provided comparison date.
  - Returns 0 if the receiver's date is later than or equal to the provided comparison date. *
```

- @param date NSDate - Provided date for comparison \*
- @return double representing the number of seconds / -  
`(double)secondsEarlierThan:(NSDate )date{ return ABS(MIN([self secondsFrom:date], 0)); }`

## pragma mark Later Than

/\*\*

- Returns the number of years the receiver's date is later than the provided comparison date.
- Returns 0 if the receiver's date is earlier than or equal to the provided comparison date. \*
- @param date NSDate - Provided date for comparison \*
- @return NSInteger representing the number of years / -  
`(NSInteger)yearsLaterThan:(NSDate )date{ return MAX([self yearsFrom:date], 0); }`

/\*\*

- Returns the number of months the receiver's date is later than the provided comparison date.
- Returns 0 if the receiver's date is earlier than or equal to the provided comparison date. \*
- @param date NSDate - Provided date for comparison \*
- @return NSInteger representing the number of months / -  
`(NSInteger)monthsLaterThan:(NSDate )date{ return MAX([self monthsFrom:date], 0); }`

/\*\*

- Returns the number of weeks the receiver's date is later than the provided comparison date.
- Returns 0 if the receiver's date is earlier than or equal to the provided comparison date. \*
- @param date NSDate - Provided date for comparison \*
- @return NSInteger representing the number of weeks / -  
`(NSInteger)weeksLaterThan:(NSDate )date{ return MAX([self`

- ```

        weeksFrom:date], 0); }

/**
• Returns the number of days the receiver's date is later than the provided
comparison date.
• Returns 0 if the receiver's date is earlier than or equal to the provided
comparison date. *
• @param date NSDate - Provided date for comparison *
• @return NSInteger representing the number of days / -
(NSUInteger)daysLaterThan:(NSDate )date{ return MAX([self daysFrom:date],
0); }

/**
• Returns the number of hours the receiver's date is later than the provided
comparison date.
• Returns 0 if the receiver's date is earlier than or equal to the provided
comparison date. *
• @param date NSDate - Provided date for comparison *
• @return double representing the number of hours / -(double)hoursLaterThan:
(NSDate )date{ return MAX([self hoursFrom:date], 0); }

/**
• Returns the number of minutes the receiver's date is later than the provided
comparison date.
• Returns 0 if the receiver's date is earlier than or equal to the provided
comparison date. *
• @param date NSDate - Provided date for comparison *
• @return double representing the number of minutes / -
(double)minutesLaterThan:(NSDate )date{ return MAX([self
minutesFrom:date], 0); }

/**
• Returns the number of seconds the receiver's date is later than the provided
comparison date.
• Returns 0 if the receiver's date is earlier than or equal to the provided
comparison date. *

```

- @param date NSDate - Provided date for comparison \*
- @return double representing the number of seconds / -
 

```
(double)secondsLaterThan:(NSDate *)date{ return MAX([self secondsFrom:date], 0); }
```

## pragma mark Comparators

```
/**
```

- Returns a YES if receiver is earlier than provided comparison date, otherwise returns NO \*
- @param date NSDate - Provided date for comparison \*
- @return BOOL representing comparison result / -(BOOL)isEarlierThan:
 

```
(NSDate *)date{ if (self.timeIntervalSince1970 < date.timeIntervalSince1970) {
```

```
    return YES;
```

```
}
```

```
/**
```

- Returns a YES if receiver is later than provided comparison date, otherwise returns NO \*
- @param date NSDate - Provided date for comparison \*
- @return BOOL representing comparison result / -(BOOL)isLaterThan:
 

```
(NSDate *)date{ if (self.timeIntervalSince1970 > date.timeIntervalSince1970) {
```

```
    return YES;
```

```
}
```

```
/**
```

- Returns a YES if receiver is earlier than or equal to the provided comparison date, otherwise returns NO \*
- @param date NSDate - Provided date for comparison \*
- @return BOOL representing comparison result / -
 

```
(BOOL)isEarlierThanOrEqualTo:(NSDate *)date{ if (self.timeIntervalSince1970
```

```

<= date.timeIntervalSince1970) {
    return YES;
}

} return NO; }

/**

```

- Returns a YES if receiver is later than or equal to provided comparison date, otherwise returns NO \*
- @param date NSDate - Provided date for comparison \*
- @return BOOL representing comparison result / -
 $(BOOL)isLaterThanOrEqualTo:(NSDate )date\{ if (self.timeIntervalSince1970 >= date.timeIntervalSince1970) {$

```

    return YES;
}
```

```

} return NO; }
```

## pragma mark - Formatted Dates

### pragma mark Formatted With Style

```

/**
```

- Convenience method that returns a formatted string representing the receiver's date formatted to a given style \*
- @param style NSDateFormatterStyle - Desired date formatting style \*
- @return NSString representing the formatted date string / -(NSString
 $)formattedDateWithStyle:(NSDateFormatterStyle)style\{ return [self
formattedDateWithStyle:style timeZone:[NSTimeZone systemTimeZone]
locale:[NSLocale autoupdatingCurrentLocale]]; \}$

```

/**
```

- Convenience method that returns a formatted string representing the receiver's date formatted to a given style and time zone \*

- @param style NSDateFormatterStyle - Desired date formatting style
  - @param timeZone NSTimeZone - Desired time zone \*
  - @return NSString representing the formatted date string / -(NSString )formattedDateWithStyle:(NSDateFormatterStyle)style timeZone:(NSTimeZone \*)timeZone{ return [self formattedDateWithStyle:style timeZone:timeZone locale:[NSLocale autoupdatingCurrentLocale]]; }
- /\*\*
- Convenience method that returns a formatted string representing the receiver's date formatted to a given style and locale \*
  - @param style NSDateFormatterStyle - Desired date formatting style
  - @param locale NSLocale - Desired locale \*
  - @return NSString representing the formatted date string / -(NSString )formattedDateWithStyle:(NSDateFormatterStyle)style locale:(NSLocale \*)locale{ return [self formattedDateWithStyle:style timeZone:[NSTimeZone systemTimeZone] locale:locale]; }
- /\*\*
- Convenience method that returns a formatted string representing the receiver's date formatted to a given style, time zone and locale \*
  - @param style NSDateFormatterStyle - Desired date formatting style
  - @param timeZone NSTimeZone - Desired time zone
  - @param locale NSLocale - Desired locale \*
  - @return NSString representing the formatted date string / -(NSString )formattedDateWithStyle:(NSDateFormatterStyle)style timeZone:(NSTimeZone )timeZone locale:(NSLocale )locale{ static NSDateFormatter \*formatter = nil; static dispatch\_once\_t onceToken; dispatch\_once(&onceToken, ^{

```
formatter = [[NSDateFormatter alloc] init];
```

});
  - [formatter setDateStyle:style]; [formatter setTimeZone:timeZone]; [formatter setLocale:locale]; return [formatter stringFromDate:self]; }

# pragma mark Formatted With Format

```
/**
```

- Convenience method that returns a formatted string representing the receiver's date formatted to a given date format \*
- @param format NSString - String representing the desired date format \*
- @return NSString representing the formatted date string / -(NSString )formattedDateWithFormat:(NSString \*)format{ return [self formattedDateWithFormat:format timeZone:[NSTimeZone systemTimeZone] locale:[NSLocale autoupdatingCurrentLocale]]; }

```
/**
```

- Convenience method that returns a formatted string representing the receiver's date formatted to a given date format and time zone \*
- @param format NSString - String representing the desired date format
- @param timeZone NSTimeZone - Desired time zone \*
- @return NSString representing the formatted date string / -(NSString )formattedDateWithFormat:(NSString )format timeZone:(NSTimeZone )timeZone{ return [self formattedDateWithFormat:format timeZone:timeZone locale:[NSLocale autoupdatingCurrentLocale]]; }

```
/**
```

- Convenience method that returns a formatted string representing the receiver's date formatted to a given date format and locale \*
- @param format NSString - String representing the desired date format
- @param locale NSLocale - Desired locale \*
- @return NSString representing the formatted date string / -(NSString )formattedDateWithFormat:(NSString )format locale:(NSLocale )locale{ return [self formattedDateWithFormat:format timeZone:[NSTimeZone systemTimeZone] locale:locale]; }

```
/**
```

- Convenience method that returns a formatted string representing the receiver's date formatted to a given date format, time zone and locale \*
- @param format NSString - String representing the desired date format

- @param timeZone NSTimeZone - Desired time zone
  - @param locale NSLocale - Desired locale \*
  - @return NSString representing the formatted date string / -(NSString \*)formattedDateWithFormat:(NSString )format timeZone:(NSTimeZone )timeZone locale:(NSLocale )locale{ static NSDateFormatter formatter = nil; static dispatch\_once\_t onceToken; dispatch\_once(&onceToken, ^{

```
formatter = [[NSDateFormatter alloc] init];
```

});
- ```
[formatter setDateFormat:format]; [formatter setTimeZone:timeZone];
[formatter setLocale:locale]; return [formatter stringFromDate:self]; }
```

## pragma mark - Helpers

```
/**
```

- Class method that returns whether the given year is a leap year for the Gregorian Calendar
- Returns YES if year is a leap year, otherwise returns NO \*
- @param year NSInteger - Year to evaluate \*
- @return BOOL evaluation of year / +(BOOL)isLeapYear:(NSInteger)year{ if (year%400){

```
    return YES;
```

```
} else if (year%100){
```

```
    return NO;
```

```
} else if (year%4){
```

```
    return YES;
```

```
    }

    return NO; }

/**

• Retrieves the default calendar identifier used for all non-calendar-specified
operations *
• @return NSString - NSCalendarIdentifier / +(NSString
)defaultCalendarIdentifier { return defaultCalendarIdentifier; }

/**

• Sets the default calendar identifier used for all non-calendar-specified
operations *
• @param identifier NSString - NSCalendarIdentifier */
    ◦ (void)setDefaultCalendarIdentifier:(NSString *)identifier {
        defaultCalendarIdentifier = [identifier copy]; implicitCalendar =
[[NSCalendar alloc] initWithCalendarIdentifier:defaultCalendarIdentifier ?: NSGregorianCalendar]; }

/**

• Retrieves a default NSCalendar instance, based on the value of
defaultCalendarSetting *
• @return NSCalendar The current implicit calendar */
    ◦ (NSCalendar *)implicitCalendar { return implicitCalendar; }

@end
```

## #####3. 时间显示格式

|                                                             |
|-------------------------------------------------------------|
| 格式   显示                                                     |
| :--:   :--:                                                 |
| dd/MM/yyyy hh:mm a   30/03/2015 4:00 PM                     |
| MMM dd,YYYY   Apr 11,2015                                   |
| EEE MMM dd HH:mm:ss Z yyyy   Thu Apr 09 14:37:22 +0800 2015 |

## #####4. 时间工具方法

```
1. 从 NSString -> NSDate
````objectivec
+ (NSDate *)getDateFromString:(NSString *)pstrDate format:(NSString *)aFormat
{
    if (pstrDate.length == 0)
    {
        return nil;
    }

    NSDateFormatter *df1 = [[NSDateFormatter alloc] init];
    [df1 setDateFormat:aFormat];
    NSDate *dtPostDate = [df1 dateFromString:pstrDate];
    return dtPostDate;
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-06-09 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

NSMutableSet - 在NSMutableSet中添加自定义对象时怎么保证不重复

## 关键字

NSMutableSet \ hash \ isEqual \ 重复

## 需求场景

1. 需要将自定义对象添加入 NSMutableSet，需要保证不重复
2. 避免存储自定义重复对象

## 参考链接

1. [NSHipster - Equality](#)

## 详细内容

通过自定义 `hash` 与 `isEqual` 方法定义相等条件

示例代码如下：

ALFImport.h

```

//  

//  ALFImport.h  

//  ALFUMLTool  

//  

//  Created by Alfred Jiang on 6/8/15.  

//  Copyright (c) 2015 Alfred Jiang. All rights reserved.  

//  
  

#import <Foundation/Foundation.h>  
  

typedef NS_ENUM(NSInteger, ImportType) {  

    IMPORT_TYPE_SYSTEM = 0,  

    IMPORT_TYPE_USER,  

    IMPORT_TYPE_WEAK  

};  
  

@interface ALFImport : NSObject  

{  

    ImportType type;  

    NSString *name;  

//    NSArray *classes;  

}  
  

@property(nonatomic,assign) ImportType type;  

@property(nonatomic,strong) NSString *name;  

//@property(nonatomic,strong) NSArray *classes;  
  

- (NSUInteger)hash;  

- (BOOL)isEqual:(id)object;  

- (NSString *)description;  
  

@end

```

## ALFImport.m

```

//  

//  ALFImport.m  

//  ALFUMLTool  

//

```

```
// Created by Alfred Jiang on 6/8/15.  
// Copyright (c) 2015 Alfred Jiang. All rights reserved.  
  
  
#import "ALFImport.h"  
  
@implementation ALFImport  
@synthesize type;  
@synthesize name;  
//@synthesize classes;  
  
- (BOOL)isEqualToImport:(ALFImport *)import {  
    if (!import) {  
        return NO;  
    }  
  
    BOOL haveEqualType = (!self.type && !import.type) || (type ==  
    import.type);  
    BOOL haveEqualName = (!self.name && !import.name) || [self.n  
ame isEqualToString:import.name];  
  
    return haveEqualType && haveEqualName;  
}  
  
- (NSUInteger)hash  
{  
    return self.type ^ [self.name hash];  
}  
  
- (BOOL)isEqual:(id)object  
{  
    if (self == object) {  
        return YES;  
    }  
  
    if (![object isKindOfClass:[ALFImport class]]) {  
        return NO;  
    }  
  
    return [self isEqualToImport:(ALFImport *)object];
```

```
}

- (NSString *)description
{
    return [NSString stringWithFormat:@"Import Name : %@", name];
}

@end
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-08-07 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

NSObject - 实现自定义对象 isEqual 方法

## 关键字

NSObject \ 自定义对象 \ 相等 \ isEqual \ Hash

## 需求场景

1. 需要对自定义对象进行相等判断时
2. 需要对自定义对象数组进行是否包含判断时

## 参考链接

1. [CSDN - OC判断对象是否相等](#)
2. [CSDN - Objective-C 实现Equality and Hashing](#)

## 详细内容

```
@implementation Person

- (BOOL)isEqual:(id)object {
    if (self == object) return YES;
    if (![object isKindOfClass:[Person class]]) return NO;

    return [self.name isEqualToString:[object name]];
}

- (NSUInteger)hash {
    return [self.name hash];
}

@end
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-07-14 | Alfred Jiang | -  |

## 方案名称

NSString - 删除 NSString 中前后空格或回车符

## 关键字

NSString \ 删除 \ 字符串 \ 首尾空格

## 需求场景

1. 删除 NSString 中的前后空格或回车符

## 参考链接

1. [NSString 去掉前后空格或回车符](#)

## 详细内容

```
NSString *string = @" spaces in front and at the end ";
NSString *trimmedString = [string stringByTrimmingCharactersInSet:[NSCharacterSet whitespaceAndNewlineCharacterSet]];
NSLog(trimmedString);
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-08-21 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

NSString - 删除 NSString 中特定字符

## 关键字

NSString \ 删除 \ 字符串

## 需求场景

1. 删除 NSString 中的数字、符号，或者修改其中的字符

## 参考链接

1. [iOS 删除 NSString 中特定字符](#)

## 详细内容

```
+ (NSString *) stringDeleteString:(NSString *)str
{
    NSMutableString *str1 = [NSMutableString stringWithString:str];
    for (int i = 0; i < str1.length; i++) {
        unichar c = [str1 characterAtIndex:i];
        NSRange range = NSMakeRange(i, 1);
        if (c == '\"' || c == '.' || c == ',' || c == '(' || c ==
')') { //此处可以是任何字符
            [str1 deleteCharactersInRange:range];
            --i;
        }
    }
    NSString *newstr = [NSString stringWithFormat:str1];
    return newstr;
}
```

## 效果 图

(无)

## 备 注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-26 | Alfred Jiang | 录入 |
| 2  | 2015-08-18 | Alfred Jiang | 更新 |
| 3  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

NSString - 汉字繁体简体相互转换的实现

## 关键字

NSString \ 简体汉字 \ 繁体汉字 \ 繁简转换

## 需求场景

1. 需要对本地工程显示文字进行繁体和简体互相转换时

## 参考链接

1. [GitHub - OBCCConvertor](#)
2. [GitHub - ChineseToPinYin](#)

## 详细内容

### 1. convertGB\_BIG

需要添加 *big5.txt* 和 *gb.txt* 文件

1. convertGB\_BIG.h ``objectivec // // convertGB\_BIG.h // myTest // // Created by sffofn on 11-8-17. // Copyright 2011 keke.com. All rights reserved. //

## import

```
@interface convertGB_BIG : NSObject { NSString _string_GB; NSString  
_string_BIG5; }  
  
@property(nonatomic, retain) NSString string_GB; @property(nonatomic, retain)  
NSString string_BIG5;  
  
-(NSString)gbToBig5:(NSString)srcString; -(NSString)big5ToGb:  
(NSString)srcString;  
  
@end
```

1. convertGB\_BIG.h // // convertGB\_BIG.h // myTest // // Created by sffofn on  
11-8-17. // Copyright 2011 keke.com. All rights reserved. //

## import

```
@interface convertGB_BIG : NSObject { NSString _string_GB; NSString  
_string_BIG5; }  
  
@property(nonatomic, retain) NSString string_GB; @property(nonatomic, retain)  
NSString string_BIG5;  
  
-(NSString)gbToBig5:(NSString)srcString; -(NSString)big5ToGb:  
(NSString)srcString;  
  
@end
```

```
2. convertGB_BIG.m  
```objectivec  
//  
// convertGB_BIG.m  
// myTest  
//  
// Created by sffofn on 11-8-17.  
// Copyright 2011 keke.com. All rights reserved.  
//  
  
#import "convertGB_BIG.h"
```

```

@implementation convertGB_BIG
@synthesize string_GB = _string_GB;
@synthesize string_BIG5 = _string_BIG5;

-(void)dealloc
{
    [_string_GB release];
    [_string_BIG5 release];

    [super dealloc];
}

-(id)init
{
    if(self = [super init])
    {
        NSError *error;
        NSString *resourcePath = [ [NSBundle mainBundle] resourcePath];
        self.string_GB = [NSString stringWithContentsOfFile:[resourcePath stringByAppendingPathComponent:@"gb.txt"]
                                                encoding:NSUTF8StringEncoding
                                                error:&error];
        self.string_BIG5 = [NSString stringWithContentsOfFile:[resourcePath stringByAppendingPathComponent:@"big5.txt"]
                                                encoding:NSUTF8StringEncoding
                                                error:&error];
    }
    return self;
}

//简体转繁体
-(NSString*)gbToBig5:(NSString*)srcString
{
    NSUInteger length = [srcString length];

```

```

        for (NSInteger i = 0; i < length; i++)
    {
        NSString *string = [srcString substringWithRange:NSMakeRange(i, 1)];
        NSRange gbRange = [_string_GB rangeOfString:string];
        if(gbRange.location != NSNotFound)
        {
            NSString *big5String = [_string_BIG5 substringWithRange:gbRange];
            srcString = [srcString stringByReplacingCharactersInRange:NSMakeRange(i, 1)
                                         withString:big5String];
        }
    }

    return srcString;
}

//繁体转简体
-(NSString*)big5ToGb:(NSString*)srcString
{
    NSInteger length = [srcString length];
    for (NSInteger i = 0; i < length; i++)
    {
        NSString *string = [srcString substringWithRange:NSMakeRange(i, 1)];
        NSRange big5Range = [_string_BIG5 rangeOfString:string];
        if(big5Range.location != NSNotFound)
        {
            NSString *gbString = [_string_GB substringWithRange:big5Range];
            srcString = [srcString stringByReplacingCharactersInRange:NSMakeRange(i, 1)
                                         withString:gbString];
        }
    }

    return srcString;
}

```

```
}
```

```
@end
```

## 2. OBCCConvertor

需要添加 *ts.tab* 文件

```
1. OBCCConvertor.h ````objectivec // // OBCCConvertor.h // // Created by Bill Cheng  
on 11/1/13. // Copyright 2013 R3 Studio. All rights reserved. // // Permission is  
hereby granted, free of charge, to any person obtaining a copy // of this  
software and associated documentation files (the "Software"), to deal // in the  
Software without restriction, including without limitation the rights // to use,  
copy, modify, merge, publish, distribute, sublicense, and/or sell // copies of  
the Software, and to permit persons to whom the Software is // furnished to  
do so, subject to the following conditions: // // The above copyright notice and  
this permission notice shall be included in // all copies or substantial portions  
of the Software. // // THE SOFTWARE IS PROVIDED "AS IS", WITHOUT  
WARRANTY OF ANY KIND, EXPRESS OR // IMPLIED, INCLUDING BUT  
NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, // FITNESS  
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT  
SHALL THE // AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR  
ANY CLAIM, DAMAGES OR OTHER // LIABILITY, WHETHER IN AN  
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, // OUT  
OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER  
DEALINGS IN // THE SOFTWARE. //
```

## import

```
@interface OBCCConvertor : NSObject{ NSMutableDictionary ts, st; }
```

```
// Create Instance for the class
```

- (OBCCConvertor\*)getInstance;

```
// Convert Traditional chinese to Simple chinese
```

- (NSString)t2s:(NSString)string;

```
// convert Simple chinese to Traditional chinese
```

- (NSString)s2t:(NSString)string;

```
@end
```

```
2. OBCCConvertor.m
````objectivec
//
//  OBCCConvertor.m
//
//  Created by Bill Cheng on 11/1/13.
//  Copyright 2013 R3 Studio. All rights reserved.
//
//  Permission is hereby granted, free of charge, to any person
obtaining a copy
//  of this software and associated documentation files (the "Software"),
//  to deal
//  in the Software without restriction, including without limitation
the rights
//  to use, copy, modify, merge, publish, distribute, sublicense,
//  and/or sell
//  copies of the Software, and to permit persons to whom the Software
is
//  furnished to do so, subject to the following conditions:
//
//  The above copyright notice and this permission notice shall
be included in
//  all copies or substantial portions of the Software.
//
//  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR
//  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
EVENT SHALL THE
//  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGE
S OR OTHER
//  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWI
```

```

SE, ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
// THE SOFTWARE.

//


#import "OBCCConvertor.h"

static OBCCConvertor * instance=nil;

@implementation OBCCConvertor

+ (OBCCConvertor*)getInstance
{
    @synchronized(self) {
        if (instance==nil) {
            instance=[[OBCCConvertor alloc] init];
        }
    }
    return instance;
}

- (id)init
{
    self = [super init];
    if (self) {
        // Initialize self.
        // Read code table from ts.tab file
        NSString *filrPath = [[NSBundle mainBundle] pathForResource:@"ts.tab" ofType:nil];
        NSString *data = [NSString stringWithContentsOfFile:filrPath encoding:NSUTF8StringEncoding error:NULL];

        // Change the NSString to Char
        NSMutableArray *chars = [[NSMutableArray alloc] initWithCapacity:[data length]];
        for (int i=0; i < [data length]; i++) {
            NSString *ichar = [NSString stringWithFormat:@"%C", [data characterAtIndex:i]];
            [chars addObject:ichar];
        }
    }
}

```

```

    }

    ts = [NSMutableDictionary new];
    st = [NSMutableDictionary new];

    for (int i = 0; i < [chars count] ; i = i + 2){
        NSString *one = [chars objectAtIndex:i];
        NSString *two = [chars objectAtIndex:(i + 1)];
        [st setObject:one forKey:two];
        [ts setObject:two forKey:one];
    }
}

return self;
}

- (NSString*)t2s:(NSString*)string
{
    NSString *result = @"";
    NSMutableArray *tmpArray = [[NSMutableArray alloc] initWithCapacity:[string length]];
    for (int i=0; i < [string length]; i++) {
        NSString *ichar = [NSString stringWithFormat:@"%C", [string characterAtIndex:i]];
        [tmpArray addObject:ichar];
    }
    for (NSString *s in tmpArray){
        if ([ts objectForKey:s]){
            result = [NSString stringWithFormat:@"%@%@", result, [ts objectForKey:s]];
        } else {
            result = [NSString stringWithFormat:@"%@%@", result, s];
        }
    }
    [tmpArray release];
    return result;
}

- (NSString*)s2t:(NSString*)string
{

```

```
NSString *result = @"";
NSMutableArray *tmpArray = [[NSMutableArray alloc] initWithCapacity:[string length]];
for (int i=0; i < [string length]; i++) {
    NSString *ichar = [NSString stringWithFormat:@"%C", [string characterAtIndex:i]];
    [tmpArray addObject:ichar];
}
for (NSString *s in tmpArray){
    if ([st objectForKey:s]){
        result = [NSString stringWithFormat:@"%@%@", result,
[st objectForKey:s]];
    } else {
        result = [NSString stringWithFormat:@"%@%@", result,
s];
    }
}
[tmpArray release];
return result;
}

@end
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-26 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

NSString - 汉字转为拼音显示的实现

## 关键字

NSString \ 汉字 \ 拼音 \ 转换

## 需求场景

1. 需要将汉字转为英文拼音的场景

## 参考链接

1. [Code4App - 汉字转拼音](#)
2. [GitHub - ChineseToPinYin](#)
3. [GitHub - POAPinyin](#)

## 详细内容

通过使用苹果类库 *CFStringTransform* 提供的方法实现汉字转拼音。通过 *kCFStringTransformMandarinLatin* 把汉字转换为中国拼音。通过 *kCFStringTransformStripDiacritics* 把中国拼音转换为英文字母。

```
- (NSString *)hanziToPinyin:(NSString *)aHanZi
{
    NSString *strResult = @"";

    if ([aHanZi length]) {
        NSMutableString *ms = [[NSMutableString alloc] initWithString:aHanZiText.text];
        if (CFStringTransform((__bridge CFMutableStringRef)ms, 0
, kCFStringTransformMandarinLatin, NO)) {
            NSLog(@"pinyin: %@", ms);
        }
        if (CFStringTransform((__bridge CFMutableStringRef)ms, 0
, kCFStringTransformStripDiacritics, NO)) {
            NSLog(@"pinyin: %@", ms);
        }
        strResult = [ms copy];
    }

    return strResult;
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

NSString - 相似度检测

## 关键字

NSString \ 相似度 \ Damerau Levenshtein

## 需求场景

1. 需要计算两个字符串相似度的需求

## 参考链接

1. OSChina - 分享一段iOS上用的字符串相似度算法实现
2. GitHub - NSString-DamerauLevenshtein

## 详细内容

1. 简单方案 ``objectivec // // NSString+Distance.m // Levenshtein // // Created by Dawen Rie on 12-6-4. // Copyright (c) 2012年 G4 Workshop. All rights reserved. //

## import "NSString+Distance.h"

```
static inline int min(int a, int b) { return a < b ? a : b; }
```

```
@implementation NSString (Distance)
```

- (float) likePercent:(NSString \*)target{ int n = self.length; int m = target.length; if (m==0) return n; if (n==0) return m;

```
//Construct a matrix, need C99 support int matrix[n+1][m+1];
memset(&matrix[0], 0, m+1); for(int i=1; i<=n; i++) {

    memset(&matrix[i], 0, m+1);
    matrix[i][0]=i;

} for(int i=1; i<=m; i++) {

    matrix[0][i]=i;

} for(int i=1;i<=n;i++) {

    unichar si = [self characterAtIndex:i-1];
    for(int j=1;j<=m;j++)
    {
        unichar dj = [target characterAtIndex:j-1];
        int cost;
        if(si==dj){
            cost=0;
        }
        else{
            cost=1;
        }
        const int above=matrix[i-1][j]+1;
        const int left=matrix[i][j-1]+1;
        const int diag=matrix[i-1][j-1]+cost;
        matrix[i][j]=min(above,min(left,diag));
    }
}

} return 100.0 - 100.0*matrix[n][m]/self.length; }
```

@end

```
2. 专业方案 (Damerau Levenshtein) [NSString-DamerauLevenshtein](https://github.com/JanX2/NSString-DamerauLevenshtein)
```objective-c
NSString *test1 = @"I love Objective-C";
NSString *test2 = @"I love Swift";

 NSLog(@"%@",[test1 similarityToString:test2] : %f,[test1 similarityToString:test2]);
 NSLog(@"%@",[test1 likePercent:test2] : %f,[test1 likePercent:test2]);
```
//2015-03-02 16:37:23.077 TestButton[40210:436933] [test1 similarityToString:test2] : 0.444444
//2015-03-02 16:37:23.078 TestButton[40210:436933] [test1 likePercent:test2] : 44.444443
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-08-25 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |
| 2  | 2016-05-09 | Alfred Jiang | -  |

## 方案名称

## 关键字

## 需求场景

1. 需要在一个长串中筛选出特定的字串，比如一句中文中筛选出金额数字
2. 筛选出一个字符串指定字符集合
3. 筛选出字符串中全部字母和数字

## 参考链接

1. [CocoaChina - 在NSString中找到数字转换成int,方法如下。so cool !](#)

## 详细内容

```

NSString *aStr = @"这是一句测试，里面包含了19.23这个数字";
NSCharacterSet* nonDigits = [[NSCharacterSet decimalDigitCharacterSet] invertedSet];
NSString *aDigits = [aStr stringByTrimmingCharactersInSet:nonDigits];
//aDigits = 19.23

```

```
NSString *unfilteredString = @"!@#$%^&*( )_+|abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";
NSCharacterSet *notAllowedChars = [[NSCharacterSet characterSetWithCharactersInString:@"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"] invertedSet];
NSString *resultString = [[unfilteredString componentsSeparatedByCharactersInSet:notAllowedChars] componentsJoinedByString:@""];
//resultString = abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-08-24 | Alfred Jiang | -  |

## 方案名称

NSTimer - 解决 NSTimer 的循环引用问题导致的内存泄漏

## 关键字

NSTimer \ 循环引用 \ Retain Cycle \ 内存泄漏

## 需求场景

1. 使用 NSTimer 造成对象无法释放，出现循环引用

## 参考链接

1. [Why's Blog - iOS 中的 NSTimer\(推荐\)](#)

## 详细内容

由于 `NSTimer` 中 `target` 对 `self` 对象的强引用，造成在 `dealloc` 方法中无法调用 `invalidate`；不调用 `invalidate` 永远不会进入 `dealloc` 方法，而不进入 `dealloc` 方法则永远不会调用 `invalidate`；为了解决这一问题，可以使用如下 `category` 中提供的方法初始化，避免了 `NSTimer` 中 `target` 对 `self` 对象的强引用，这样就可以在 `dealloc` 方法中正常调用 `invalidate` 了

`NSTimer+Weak.h`

```
#import <Foundation/Foundation.h>

@interface NSTimer (Weak)

+ (NSTimer *)scheduledTimerWithTimeInterval:(NSTimeInterval)interval
                                         block:(void(^)(()))block
                                         repeats:(BOOL)repeats;

+ (NSTimer *)timerWithTimeInterval:(NSTimeInterval)interval
                               block:(void(^)(()))block
                               repeats:(BOOL)repeats;

@end
```

NSTimer+Weak.m

```

#import "NSTimer+Weak.h"

@implementation NSTimer (Weak)

+ (NSTimer *)scheduledTimerWithTimeInterval:(NSTimeInterval)interval
                                         block:(void(^)(()))block
                                         repeats:(BOOL)repeats
{
    return [self scheduledTimerWithTimeInterval:interval
                                             target:self
                                             selector:@selector(timerUpdateWithBlock:)];
}

+ (NSTimer *)timerWithTimeInterval:(NSTimeInterval)interval
                               block:(void(^)(()))block
                               repeats:(BOOL)repeats
{
    return [self timerWithTimeInterval:interval
                                             target:self
                                             selector:@selector(timerUpdateWithBlock:)];
}

+ (void)timerUpdateWithBlock:(NSTimer*)timer{
    void (^timerBlock)() = timer.userInfo;
    if(timerBlock){
        timerBlock();
    }
}

@end

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-18 | Alfred Jiang | -  |

## 方案名称

NSURL - 特殊字符处理

## 关键字

NSURL \ 特殊字符 \ 参数 \ 请求

## 需求场景

- 针对某些带有特殊字符的 URL 无法正确请求数据的场景

## 参考链接

- 简书 - iOSURL特殊字符处理
- 博客园 - ios中对url中得特殊字符处理

## 详细内容

### iOS 9 之前

方法一 : **CFURLCreateStringByAddingPercentEscapes**

```

NSString *UrlEncodedString(NSString *sourceText)
{
    NSString *result = (NSString *)CFBridgingRelease(CFURLCreate
StringByAddingPercentEscapes(kCFAllocatorDefault,(CFStringRef)so
urceText ,NULL ,CFSTR("!*'();@&+$%#[ ]") ,kCFStringEncodingUTF8)
); // ", / : =
    return result;
}

```

使用

```

NSString *str = @"http://img5.imgtn.bdimg.com/it/u=1478257864,28
82073929%26fm=21%26gp=0.jpg";

NSString *urlStr = UrlEncodedString(str);

```

方法二：***stringByAddingPercentEscapesUsingEncoding***

```

NSString *str = @"http://img5.imgtn.bdimg.com/it/u=1478257864,28
82073929%26fm=21%26gp=0.jpg";

NSString *urlStr = [str stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];

```

iOS 9 之后

使用 ***stringByAddingPercentEncodingWithAllowedCharacters***

```

NSString *str = @"http://img5.imgtn.bdimg.com/it/u=1478257864,28
82073929%26fm=21%26gp=0.jpg";

NSString *urlStr = [str stringByAddingPercentEncodingWithAllowed
Characters:[NSCharacterSet URLQueryAllowedCharacterSet]];

```

以上三种方法转换结果都是一样的：

```
(lldb) po urlStr  
http://img5.imgtn.bdimg.com/it/u=1478257864,2882073929%2526fm=21  
%2526gp=0.jpg
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-05-05 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

openURL - iOS App 跳转 App Store 下载和 App Store 评论

## 关键字

openURL \ App Store \ 下载 \ 评论

## 需求场景

1. 需要进行 iOS 应用下载和应用评论场景

## 参考链接

1. [Presenting, Appirater](#)

## 详细内容

| 版本      | 跳转 <b>App Store</b> 下载                                                                  |                                                             |
|---------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------|
| iOS 7 - | <a href="http://itunes.apple.com/app/idAPP_ID">http://itunes.apple.com/app/idAPP_ID</a> | itms-apps://ax.itunes.apple.com/?type=                      |
| iOS 7 + | <a href="http://itunes.apple.com/app/idAPP_ID">http://itunes.apple.com/app/idAPP_ID</a> | itms-ap                                                     |
| iOS 8 + | <a href="http://itunes.apple.com/app/idAPP_ID">http://itunes.apple.com/app/idAPP_ID</a> | itms-apps://itunes.apple.com/?id=APP_ID&onlyLatestVersion=t |

```
NSString *templateReviewURLiOS7 = @"itms-apps://itunes.apple.com/app/idAPP_ID";
NSString *templateReviewURLiOS8 = @"itms-apps://itunes.apple.com/WebObjects/MZStore.woa/wa/viewContentsUserReviews?id=APP_ID&onlyLatestVersion=true&pageNumber=0&sortOrdering=1&type=Purple+Software";

NSString *reviewURL;
if ([[UIDevice currentDevice] systemVersion] floatValue] >= 7.0
    && [[UIDevice currentDevice] systemVersion] floatValue] < 7.1)
{
    reviewURL = [templateReviewURLiOS7 stringByReplacingOccurrencesOfString:@"APP_ID" withString:[NSString stringWithFormat:@"%d", APPIRATER_APP_ID]];
}
// iOS 8 needs a different templateReviewURL also @see https://github.com/arashpayan/appirater/issues/182
else if ([[UIDevice currentDevice] systemVersion] floatValue] >= 8.0)
{
    reviewURL = [templateReviewURLiOS8 stringByReplacingOccurrencesOfString:@"APP_ID" withString:[NSString stringWithFormat:@"%d", APPIRATER_APP_ID]];
}

[[UIApplication sharedApplication] openURL:[NSURL URLWithString:reviewURL]];
```

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-09-09 | Alfred Jiang | -  |

## 方案名称

Runtime - 使用 Aspects 实现 Method Swizzling 和 AOP 实践

## 关键字

Runtime \ Method Swizzling \ Aspect Oriented Programming \ 面向切面编程

## 需求场景

- 利用 Method Swizzling 对系统方法进行替换和追加行为

## 参考链接

- 顾鹏 - Method Swizzling 和 AOP 实践(推荐)
- GitHub - steipete/Aspects

## 详细内容

在 Objective-C 中，Aspect Oriented Programming (面向切面编程) 就是利用 Runtime 特性给指定的方法添加自定义代码。有很多方式可以实现 AOP，Method Swizzling 就是其中之一。而且幸运的是，目前已经有一些第三方库可以让你不需要了解 Runtime，就能直接开始使用 AOP。

Aspects 就是一个不错的 AOP 库，封装了 Runtime，Method Swizzling 这些黑色技巧，只提供两个简单的 API：

```
+ (id<AspectToken>)aspect_hookSelector:(SEL)selector
    withOptions:(AspectOptions)options
    usingBlock:(id)block
    error:(NSError ***)error;

- (id<AspectToken>)aspect_hookSelector:(SEL)selector
    withOptions:(AspectOptions)options
    usingBlock:(id)block
    error:(NSError ***)error;
```

示例：用户看到某个 **View** 的时候，把这个事件记下来

### 1. 通过 Runtime 提供的 Method Swizzling 方法实现

```
@implementation UIViewController (Logging)

void swizzleMethod(Class class, SEL originalSelector, SEL swizzledSelector)
{
    // the method might not exist in the class, but in its super
    // class
    Method originalMethod = class_getInstanceMethod(class, originalSelector);
    Method swizzledMethod = class_getInstanceMethod(class, swizzledSelector);

    // class_addMethod will fail if original method already exists
    BOOL didAddMethod = class_addMethod(class, originalSelector,
        method_getImplementation(swizzledMethod), method_getTypeEncoding(swizzledMethod));

    // the method doesn't exist and we just added one
    if (didAddMethod) {
        class_replaceMethod(class, swizzledSelector, method_getImplementation(originalMethod),
            method_getTypeEncoding(originalMethod));
    }
}
```

```

    }
    else {
        method_exchangeImplementations(originalMethod, swizzledMethod);
    }
}

+ (void)load
{
    swizzleMethod([self class], @selector(viewDidAppear:), @selector(swizzled_viewDidAppear:));
}

- (void)swizzled_viewDidAppear:(BOOL)animated
{
    // call original implementation
    [self swizzled_viewDidAppear:animated];

    // Logging
    [Logging logWithEventName:NSStringFromClass([self class])];
}

@end

```

要先尝试添加原 selector 是为了做一层保护，因为如果这个类没有实现 originalSelector，但其父类实现了，那 class\_getInstanceMethod 会返回父类的方法。这样 method\_exchangeImplementations 替换的是父类的那个方法，这当然不是你想要的。所以我们先尝试添加 orginalSelector，如果已经存在，再用 method\_exchangeImplementations 把原方法的实现跟新的方法实现给交换掉

## 2. 通过 Runtime 提供的替换 IMP 方法直接用新的 IMP 取代原 IMP 来实现

```
void (*gOriginalViewDidAppear) (id, SEL, BOOL);

@implementation UIViewController (Logging)

void newViewDidAppear(UIViewController *self, SEL _cmd, BOOL animated)
{
    // call original implementation
    gOriginalViewDidAppear(self, _cmd, animated);

    // Logging
    [Logging logWithEventName:[NSStringFromClass([self class])]];
}

+ (void)load
{
    Method originalMethod = class_getInstanceMethod(self, @selector(viewDidAppear:));
    gOriginalViewDidAppear = (void *)method_getImplementation(originalMethod);

    if(!class_addMethod(self, @selector(viewDidAppear:), (IMP) newViewDidAppear, method_getTypeEncoding(originalMethod))) {
        method_setImplementation(originalMethod, (IMP) newViewDidAppear);
    }
}

@end
```

## 2. 通过 **Aspects** 提供接口实现

```
@implementation UIViewController (Logging)

+ (void)load
{
    [UIViewController aspect_hookSelector:@selector(viewDidAppear:)
        withOptions:AspectPositionAfter
        usingBlock:^(id<AspectInfo> aspectInfo) {

        NSString *className = NSStringFromClass([[aspectInfo instance] class]);
        [Logging logWithEventName:className];
    }
    error:NULL];
}

@end
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-31 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UDID - 解决方案介绍与比较

## 关键字

UDID \ UUID \ 设备唯一标识符

## 需求场景

1. 需要设备唯一标识符的场景

## 参考链接

1. [CSDN - IOS7开发～UDID解决方法](#)
2. [现有IOS设备唯一标示符的方案比较](#)

## 详细内容

### 1. UDID

```
[[UIDevice currentDevice] uniqueIdentifier]
```

iOS 官方最早提供的 UDID 方案，根据某一公式，使用设备序列号、网卡地址等信息作为参数计算而来，iOS6 之后该计算公式发生了改变。

该方法返回的结果在所有应用中都相同，并且卸载应用、刷机、还原设备均不会发生改变，是最为准确的设备唯一标示符。

iOS5 之后，该方法被标记为废弃！最终，在 2013 年 5 月 1 号之后，AppStore 禁止任何使用该方法的应用上架。

iOS7 中对外公开的 API 中已经移除了该方法！

## 2. 网卡地址

通过 Unix 级别的 API 去获取 Wifi 的网卡地址。

该方法属于 UDID 替代方案中最为准确的一种，因为网卡地址不会因为刷机、还原设备而发生改变。故追求唯一性的厂商多选择此种方案！

在 iOS7 之后，应用已经无法获取网卡地址，所以该方案在 iOS7 中也已经被废弃。

## 3. OpenUDID

开源的一个 UDID 替代方案，原理是利用应用间的剪贴板共享和本地一些必要的缓存信息，让多个应用间共享一个 UUID。

OpenUDID 在官方废弃 UDID 接口之后，受到广泛的欢迎！可以说是现在大多数应用的 UDID 替代方法。

OpenUDID 在刷机、还原设备后就会产生新的 UDID，事实上，由于剪贴板的特殊性，如果所有使用了 OpenUDID 的应用被全部卸载之后，再次安装的应用取到的 OpenUDID 将会是一个全新的值！

iOS7 中，不同组的应用（即不同厂商）的应用之间不再能共享剪贴板间的数据！

同组（即同一厂商）应用的定义为：Info.plist 中关于软件唯一标识符的字段 CFBundleIdentifier 中的前两段标识符（例如 com.mycompany）相同。

固在 iOS7 中，OpenUDID 也将慢慢失去它的意义。

## 4. 保存在 NSUserDefaults 中的 UUID

在 iOS5 将 UDID 标为废弃之后，官方提供的替代方案。即使用 CFUUIDCreate 生成一个 UUID，并将之保存在 UserDefaults 中，用它作为设备标识符。在 iOS6 之后，苹果更推出 NSUUID 来替代 CFUUIDCreate，但本质是一样的。

UUID 每次都会生成一个新的字符串，也就是说应用被卸载之后，就会被认为是一个新的设备，更不用提刷机、还原设备了。

故基本无人采用 UUID 的方案。

## 5. 厂商唯一标识符 **identifierForVendor**

```
[[UIDevice currentDevice] identifierForVendor]
```

iOS6 中推出的 UDID 替代方案，该方法对于同一厂商的应用返回相同的值，不同厂商所得到的值不同。

该方案刷机、还原设备后，获得值将会改变。同样注意的是：如果同一厂商安装的所有应用都被卸载后，新安装的同一厂商的应用同样也将获得新的值，而不是原来的值！

由于不能跨厂商，并且软件卸载后再安装有改变的可能性，该方案也并没有被广大开发商接受。

## 6. 广告标识符 **advertisingIdentifier** (推荐方案)

```
[[ASIIdentifierManager sharedManager] advertisingIdentifier];
```

iOS6 中推出的另一款 UDID 替代方案，该方法对所有厂商的应用返回相同的值。同样提供的是另一个 API，`advertisingTrackingEnabled`，该参数表示用户是否希望广告追踪被限定，但该参数仅仅是个布尔值，用于表示用户意愿，不影响 `advertisingIdentifier` 的读取。

该方法由于是官方提供的，并且所有厂商的应用取到的值相同，所以相对接受度好些，但是由于仅在 iOS6 上适用，所以更多人还是选择了 OpenUDID 的方案。

该方法刷机、还原设备后，获得的值将会改变。此外，用户如果通过 设置->关于本机->广告->还原广告标识符，就可以重新生成一个新的值。

该方法是 iOS7 上目前官方允许的范畴内最为可接受的方案。

## 7. IMEI

iOS 官方 API 无法获得 IMEI，或者说禁止获取。故很少有人在 iOS 上去读取 IMEI。但 IMEI 作为设备唯一标识符是最为准确的方法之一！不会随着刷机、还原设备而改变！

读取 IMEI 的应用将会被 AppStore 拒绝！

同样类似的方案有蓝牙地址、 iOS 设备序列号（SerialNumber）等等。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注                   |
|----|------------|--------------|----------------------|
| 1  | 2015-04-22 | Alfred Jiang | -                    |
| 2  | 2015-05-11 | Alfred Jiang | 增加 RKNotificationHub |
| 3  | 2015-12-23 | Alfred Jiang | -                    |

## 方案名称

UIButton - badge 显示的实现(使用 UIBarButtonItem-Badge)

## 关键字

UIButton \ badge \ UIBarButtonItem

## 需求场景

1. 需要在按钮左上角显示未读数目

## 参考链接

1. [CocoaControls Badges](#)
2. [GitHub - UIBarButtonItem-Badge](#)
3. [GitHub - RKNotificationHub](#)

## 详细内容

### 1. 引入头文件

```
#import "UIBarButtonItem+Badge.h"
```

### 2. 添加 UIBarButtonItem

```
UIImage *image = [UIImage imageNamed:@"someImage"];
UIBarButtonItem *navLeftButton = [[UIBarButtonItem alloc] initWithImage:image
style:UIBarButtonItemStylePlain
target:self
action:@selector(buttonPress:)];
self.navigationItem.leftBarButtonItem = navLeftButton;
self.navigationItem.leftBarButtonItem.badgeValue = @"1";
```

### 3. 有用的属性

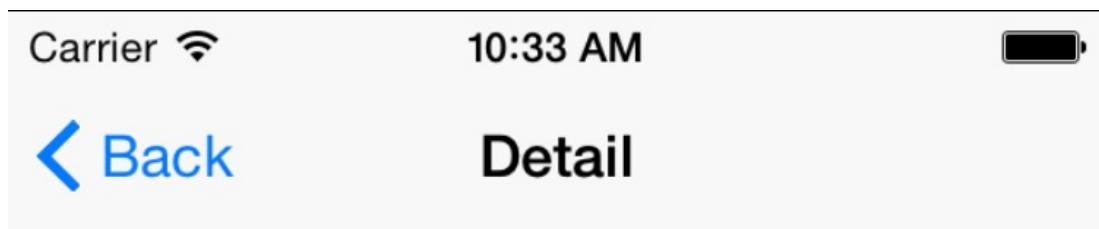
```
// Each time you change one of properties, the badge will refresh with your changes

// Badge value to be display
@property (nonatomic) NSString *badgeValue;
// Badge background color
@property (nonatomic) UIColor *badgeBackgroundColor;
// Badge text color
@property (nonatomic) UIColor *badgeTextColor;
// Badge font
@property (nonatomic) UIFont *badgeFont;

// Padding value for the badge
@property (nonatomic) CGFloat badgePadding;
// Minimum size badge to small
@property (nonatomic) CGFloat badgeMinSize;
// Values for offseting the badge over the BarButtonItem you picked
@property (nonatomic) CGFloat badgeOriginX;
@property (nonatomic) CGFloat badgeOriginY;

// In case of numbers, remove the badge when reaching zero
@property BOOL shouldHideBadgeAtZero;
// Badge has a bounce animation when value changes
@property BOOL shouldAnimateBadge;
```

## 效果图



2014-10-09 14:30:17 +0000



## 备注

当嵌套 *SlideNavigationController* 等控件时，需要注意 *leftBarButtonItem* 的对象究竟属于哪个 *UIViewController*

```
func badgeValue(aValue : NSString)
{
    var anewValue = aValue.integerValue > 99 ? "99+" : aValue
    self.navigationItem.leftBarButtonItem?.badgeValue = anewValue
}
```

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-02 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UIButton - 使用 RNLoadingButton 实现等待按钮

## 关键字

UIButton \ 等待 \ 按钮

## 需求场景

1. 实现一个显示等待状态的按钮

## 参考链接

1. [GitHub - RNLoadingButton-Swift](#)
2. [GitHub - RNLoadingButton](#)

## 详细内容

用法示例

```
//Mark: Can usage with Nib
// Configure State
btn1.hideTextWhenLoading = false
btn1.loading = false
btn1.activityIndicatorAlignment = RNActivityIndicatorAlignment.Right
btn1.activityIndicatorEdgeInsets = UIEdgeInsetsMake(0, 50, 0, 10)
)
btn1.setTitle("connecting", forState: UIControlState.Disabled)

//等待完成
//(sender: RNLoadingButton)

sender.enabled = false
sender.loading = true
```

## 效果图

(无)

## 备注

(无)

## n### 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-01 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UIButton - 避免多次重复点击

## 关键字

UIButton \ 多次重复点击 \ 延时操作

## 需求场景

- 避免短时间内多次重复点击按钮造成的响应错误

## 参考链接

(无)

## 详细内容

- Swift 解决方案 ``swift // // UIButtonExtension.swift // GrandJustice // //  
Created by Alfred Jiang on 1/29/15. // Copyright (c) 2015 FYH. All rights reserved. //

import Foundation

```
extension UIButton { func antiMultiplyTouch(delay : NSTimeInterval, closure:()->())  
{ self.userInteractionEnabled = false
```

```

        dispatch_after(
            dispatch_time(
                DISPATCH_TIME_NOW,
                Int64(delay * Double(NSEC_PER_SEC))
            ),
            dispatch_get_main_queue(), {
                self.userInteractionEnabled = true
                closure()
            }
        )
    }
}

```

## 2. Objective-C 解决方案

```

```objectivec
== .h文件
//
//  UIButton+AntiMultiplyTouch.h
//  TestButton
//
//  Created by Alfred Jiang on 3/1/15.
//  Copyright (c) 2015 Alfred Jiang. All rights reserved.
//


#import <UIKit/UIKit.h>

@interface UIButton (antiMultiplyTouch)

- (void)antiMultiplyTouch:(NSTimeInterval)delay block:(void(^)(void))operation;

@end

== .m文件
#import "UIButton+AntiMultiplyTouch.h"

@implementation UIButton (AntiMultiplyTouch)

```

```
- (void)antiMultiplyTouch:(NSTimeInterval)delay block:(void(^)(void))operation
{
    self.userInteractionEnabled = NO;

    dispatch_after(
        dispatch_time(
            DISPATCH_TIME_NOW,
            delay * NSEC_PER_SEC
        ),
        dispatch_get_main_queue(),
        ^{
            self.userInteractionEnabled = YES;
            operation();
        }
    );
}

@end
```

## 效果图

(无)

## 备注

OC方案暂未测试

- [UIView \ UIButton - 独占响应事件](#)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-10-14 | Alfred Jiang | -  |

## 方案名称

UIDevice - 通过 DeviceKit 更方便的获取硬件设备参数

## 关键字

UIDevice \ DeviceKit \ 设备 \ 模拟器 \ 电量信息与状态

## 需求场景

1. 需要确认设备型号时
2. 需要确认电量信息状态时

## 参考链接

1. [GitHub - dennisweissmann/DeviceKit\(推荐\)](#)

## 详细内容

### Usage

Here are some usage examples. All devices are also available as simulators:

```
.iPhone6 => .Simulator(.iPhone6)
.iPhone6s => .Simualtor(.iPhone6s)
```

etc.

**Get the Device You're Running On**

```
let device = Device()

print(device)      // prints, for example, "iPhone 6 Plus"

if device == .iPhone6Plus {
    // Do something
} else {
    // Do something else
}
```

## Get the Device Family

```
let device = Device()
if device.isPod {
    // iPods (real or simulator)
} else if device.isPhone {
    // iPhone (real or simulator)
} else if device.isPad {
    // iPad (real or simulator)
}
```

## To check if running on Simulator

```
let device = Device()
if device.isSimulator {
    // Running on one of the simulators(iPod/iPhone/iPad)
    // Skip doing something irrelevant for Simulator
}
```

## Get the Simulator Device

```

let device = Device()
switch device {
    case .Simulator(.iPhone6s): break // You're running on the iPhone 6s simulator
    case .Simulator(.iPadAir2): break // You're running on the iPad Air 2 simulator
    default: break
}

```

## Make Sure the Device Is Contained in a Preconfigured Group

```

let groupOfAllowedDevices: [Device] = [.iPhone6, .iPhone6Plus, .iPhone6s, .iPhone6sPlus, .Simulator(.iPhone6), .Simulator(.iPhone6Plus), .Simulator(.iPhone6s), .Simulator(.iPhone6sPlus)]
let device = Device()

if device.isOneOf(groupOfAllowedDevices) {
    // Do your action
}

```

## Get the Current Battery State

```

if device.batteryState == .Full || device.batteryState >= .Charging(75) {
    print("Your battery is happy! 😊")
}

```

## Get the Current Battery Level

```

if device.batteryLevel >= 50 {
    install_iOS()
} else {
    showError()
}

```

效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-04-14 | Alfred Jiang | -  |

## 方案名称

UIEvent - iOS 事件拦截

## 关键字

UIEvent \ 事件拦截 \ 捕获任意点击事件

## 需求场景

1. 需要响应任意点击事件
2. 需要接收任意交互事件信息

## 参考链接

1. 博客园 - iOS 事件拦截(实现触摸任意位置隐藏指定view)

## 详细内容

1. 新建一个自定义的 **MyApplication** 继承自 **UIApplication**，在 **MyApplication** 中实现 **-(void)sendEvent:(UIEvent \*)event\*** 方法

```

- (void)sendEvent:(UIEvent *)event
{
    //接收 event 并发送通知
    [[NSNotificationCenter defaultCenter] postNotificationName:N
otificationMyApplicationSendEvent object:event];
    [super sendEvent:event];
}

```

## 2. 修改 main.m 中 *int main(int argc, char argv[])\** 方法

```
int main(int argc, char *argv[])
{
    @autoreleasepool {
        return UIApplicationMain(argc, argv, NSStringFromClass([MyApplication class]), NSStringFromClass([AppDelegate class]));
    }
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-22 | Alfred Jiang | -  |

## 方案名称

UIImage - 获取 UIImage 大小与压缩 UIImage

## 关键字

UIImage \ 大小 \ 尺寸 \ 压缩

## 需求场景

1. 需要对加载的 UIImage 进行压缩时

## 参考链接

1. CSDN - UIImage 图片处理：截图，缩放，设定大小，存储
2. Stack Overflow - iOS get UIImage memory size

## 详细内容

### 获取 **UIImage** 大小

```
UIImage *testImage = [UIImage imageNamed:@"blue2.png"]; //877KB  
600*600像素

NSUInteger s1 = UIImagePNGRepresentation(testImage).length;
//992400

NSUInteger s2 = UIImageJPEGRepresentation(testImage, 1).length;
//923162

NSUInteger s3 = CGImageGetHeight(testImage.CGImage) * CGImageGetBytesPerRow(testImage.CGImage); //1440000 = 600 * 600 * 4
```

测试文件大小为 877KB 600\*600像素

其中 s1 为保存 PNG 格式文件最大内存占用大小

其中 s2 为保存 JPEG 格式文件最大内存占用大小

其中 s3 为该图片文件当前实际内存占用大小，与像素有关

从测试数据可见实际使用图片所占用内存大小往往大于图片实际大小，所以建议对较大图片进行压缩使用

## 压缩 UIImage

UIImage+Scale.h

```
@interface UIImage (Scale)

//等比率缩放
- (UIImage *)scaleImageToScale:(float)scale;

//自定长宽
- (UIImage *)scaleImageToSize:(CGSize)size;

@end
```

UIImage+Scale.m

```
#import "UIImage+Scale.h"

@implementation UIImage (Scale)

//等比率缩放
- (UIImage *)scaleImageToScale:(float)scale
{
    UIGraphicsBeginImageContext(CGSizeMake(self.size.width * scale, self.size.height * scale));
    [self drawInRect:CGRectMake(0, 0, self.size.width * scale, self.size.height * scale)];
    UIImage *scaledImage = UIGraphicsGetImageFromCurrentImageContext();
    UIGraphicsEndImageContext();
    return scaledImage;
}

//自定长宽
- (UIImage *)scaleImageToSize:(CGSize)size
{
    UIGraphicsBeginImageContext(size);
    [self drawInRect:CGRectMake(0, 0, size.width, size.height)];
    UIImage * scaledImage = UIGraphicsGetImageFromCurrentImageContext();
    UIGraphicsEndImageContext();
    return scaledImage;
}

@end
```

使用

```
- (void)imagePickerController:(UIImagePickerController *)picker
didFinishPickingMediaWithInfo:(NSDictionary *)info
{
    UIImage *imagePhoto = [info objectForKey:UIImagePickerControllerOriginalImage];

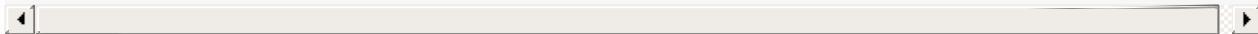
    NSData * imageData_before = UIImageJPEGRepresentation(imagePhoto, 1);
    NSLog(@"Before : %d , %@", [imageData_before length]/1024, imagePhoto);

    imagePhoto = [imagePhoto scaleImageToSize:CGSizeMake(600, 600)];
}

NSData * imageData_after = UIImageJPEGRepresentation(imagePhoto, 1);
NSLog(@"After : %d , %@", [imageData_after length]/1024, imagePhoto);
}
```

输出：

```
Before : 1504 , <UIImage: 0x796561e0> size {2560, 1600} orientation 0 scale 1.000000
After : 341 , <UIImage: 0x78e4e2f0> size {600, 600} orientation 0 scale 1.000000
```



## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-08-17 | Alfred Jiang | -  |
| 3  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UIImageView - 使用 LBB blurredImage 实现图像模糊效果

## 关键字

UIImageView \ 图像模糊 \ LBB blurredImage \ UIImageView \ 模糊 \ 半透明 \ 毛玻璃 \ 高斯模糊

## 需求场景

1. 实现页面或图片的模糊效果

## 参考链接

1. GitHub - LBB blurredImage
2. 知乎 - iOS 7 的实时毛玻璃模糊 (live blur) 效果渲染需要多大的系统开销？
3. 三石·道 - iOS 开发使用半透明模糊效果

## 详细内容

### 使用 **LBB blurredImage** 实现

1. 在工程中添加 **UIImageView+LBB blurredImage.{h,m}** 和 **UIImage+ImageEffects.{h,m}** 文件
2. 在需要实现图像模糊的类中引入 **UIImageView+LBB blurredImage.h** 文件

### 3. 使用下面的代码实现图像模糊

```
[self.imageView setImageToBlur:[UIImage imageNamed:@"example"]
    blurRadius:kLBBlurredImageDefaultBlurRadius
    completionBlock:^{
        NSLog(@"The blurred image has been set");
    }];
}
```

## 高斯模糊 Accelerate 实现

- 真实时高斯模糊貌似是不可能了（流畅的）；对于局部动态的高斯模糊：
- CoreImage 的帧数会随着模糊大小而降低，cpu也蛮高；
- Accelerate 的模糊是最靠谱的，速度和效果都非常完美，同时会占用更高的 cpu；
- GPUImage 的速度和效果都不好，但是cpu占用更少（我想一定是哪里用错了才得出这个结论的，因为所有人都说GPUImage碉堡了，秒杀CoreImage）；所以真要在项目里使用毛玻璃效果的话好像就只能在局部做个伪实时的效果吧。

### Accelerate的模糊方法

```
- (UIImage *)accelerateBlurWithImage:(UIImage *)image
{
    NSInteger boxSize = (NSInteger)(10 * 5);
    boxSize = boxSize - (boxSize % 2) + 1;

    CGImageRef img = image.CGImage;

    vImage_Buffer inBuffer, outBuffer, rgbOutBuffer;
    vImage_Error error;

    void *pixelBuffer, *convertBuffer;

    CGDataProviderRef inProvider = CGImageGetDataProvider(img);
    CFDataRef inBitmapData = CGDataProviderCopyData(inProvider);

    convertBuffer = malloc( CGImageGetBytesPerRow(img) * CGImage
GetHeight(img) );
    rgbOutBuffer.width = CGImageGetWidth(img);
    rgbOutBuffer.height = CGImageGetHeight(img);
```

```

rgbOutBuffer.rowBytes = CGImageGetBytesPerRow(img);
rgbOutBuffer.data = convertBuffer;

inBuffer.width = CGImageGetWidth(img);
inBuffer.height = CGImageGetHeight(img);
inBuffer.rowBytes = CGImageGetBytesPerRow(img);
inBuffer.data = (void *)CFDataGetBytePtr(inBitmapData);

pixelBuffer = malloc( CGImageGetBytesPerRow(img) * CGImageGetHeight(img) );

if (pixelBuffer == NULL) {
    NSLog(@"No pixelbuffer");
}

outBuffer.data = pixelBuffer;
outBuffer.width = CGImageGetWidth(img);
outBuffer.height = CGImageGetHeight(img);
outBuffer.rowBytes = CGImageGetBytesPerRow(img);

void *rgbConvertBuffer = malloc( CGImageGetBytesPerRow(img)
* CGImageGetHeight(img) );
vImage_Buffer outRGBBuffer;
outRGBBuffer.width = CGImageGetWidth(img);
outRGBBuffer.height = CGImageGetHeight(img);
outRGBBuffer.rowBytes = 3;
outRGBBuffer.data = rgbConvertBuffer;

error = vImageBoxConvolve_ARGB8888(&inBuffer, &outBuffer, NULL,
0, 0, boxSize, boxSize, NULL, kvImageEdgeExtend);

if (error) {
    NSLog(@"error from convolution %ld", error);
}
const uint8_t mask[] = {2, 1, 0, 3};

vImagePermuteChannels_ARGB8888(&outBuffer, &rgbOutBuffer, mask,
kvImageNoFlags);

CGColorSpaceRef colorSpace = CGColorSpaceCreateDeviceRGB();

```

```
CGContextRef ctx = CGBitmapContextCreate(rgbOutBuffer.data,
                                         rgbOutBuffer.width,
                                         rgbOutBuffer.height

                                         ,
                                         8,
                                         rgbOutBuffer.rowBytes,
                                         colorSpace,
                                         kCGImageAlphaNoneSkipLast);

    CGImageRef imageRef = CGBitmapContextCreateImage(ctx);
    UIImage *returnImage = [UIImage imageWithCGImage:imageRef];

    //clean up
    CGContextRelease(ctx);

    free(pixelBuffer);
    free(convertBuffer);
    free(rgbConvertBuffer);
    CFRelease(inBitmapData);

    CGColorSpaceRelease(colorSpace);
    CGImageRelease(imageRef);

    return returnImage;
}
```

## 效果图

(无)

## 备注

苹果官方示例

- [Apple documentation - Next Blurring and Tinting an Image](#)

UIImageEffects.h

```
/*
 File: UIImageEffects.h
 Abstract: This class contains methods to apply blur and tint effects to an image.
 This is the code you'll want to look out to find out how to use vImage to
 efficiently calculate a blur.
 Version: 1.1

 Disclaimer: IMPORTANT: This Apple software is supplied to you by Apple Inc. ("Apple") in consideration of your agreement to the following terms, and your use, installation, modification or redistribution of this Apple software constitutes acceptance of these terms. If you do not agree with these terms, please do not use, install, modify or redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and subject to these terms, Apple grants you a personal, non-exclusive license, under Apple's copyrights in this original Apple software (the "Apple Software"), to use, reproduce, modify and redistribute the Apple Software, with or without modifications, in source and/or binary forms; provided that if you redistribute the Apple Software in its entirety and without modifications, you must retain this notice and the following text and disclaimers in all such redistributions of the Apple Software. Neither the name, trademarks, service marks or logos of Apple Inc. may
```

be used to endorse or promote products derived from the Apple Software without specific prior written permission from Apple. Except as expressly stated in this notice, no other rights or licenses, express or implied, are granted by Apple herein, including but not limited to any patent rights that may be infringed by your derivative works or by other works in which the Apple Software may be incorporated.

The Apple Software is provided by Apple on an "AS IS" basis. A

PPLE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION

THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS

FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND

OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL

OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED

AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),

STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE

POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2014 Apple Inc. All Rights Reserved.

\*/

```
#import <Foundation/Foundation.h>

@interface UIImageEffects : NSObject

+ (UIImage*)imageByApplyingLightEffectToImage:(UIImage*)inputImage;
+ (UIImage*)imageByApplyingExtraLightEffectToImage:(UIImage*)inputImage;
+ (UIImage*)imageByApplyingDarkEffectToImage:(UIImage*)inputImage;
+ (UIImage*)imageByApplyingTintEffectWithColor:(UIColor *)tintColor toImage:(UIImage*)inputImage;

//| -----
//!
//! Applies a blur, tint color, and saturation adjustment to @a
//! inputImage,
//! optionally within the area specified by @a maskImage.
//!
//! @param inputImage
//!       The source image. A modified copy of this image will
//! be returned.
//! @param blurRadius
//!       The radius of the blur in points.
//! @param tintColor
//!       An optional UIColor object that is uniformly blended
//! with the
//!       result of the blur and saturation operations. The alpha
//! channel
//!       of this color determines how strong the tint is.
//! @param saturationDeltaFactor
//!       A value of 1.0 produces no change in the resulting image.
//!       Values less than 1.0 will desaturate the resulting image
//! while values
//!       greater than 1.0 will have the opposite effect.
//! @param maskImage
//!       If specified, @a inputImage is only modified in the
//! area(s) defined
```

```
///      by this mask. This must be an image mask or it must
///      meet the
///      requirements of the mask parameter of CGContextClipToMask.
+ (UIImage *)imageByApplyingBlurToImage:(UIImage *)inputImage with
Radius:(CGFloat)blurRadius tintColor:(UIColor *)tintColor saturationDeltaFactor:(CGFloat)saturationDeltaFactor maskImage:(UIImage *)maskImage;

@end
```

## UIImageEffects.m

```
/*
File: UIImageEffects.m
Abstract: This class contains methods to apply blur and tint effects to an image.
This is the code you'll want to look out to find out how to use vImage to
efficiently calculate a blur.
Version: 1.1

Disclaimer: IMPORTANT: This Apple software is supplied to you by Apple Inc. ("Apple") in consideration of your agreement to the following terms, and your use, installation, modification or redistribution of this Apple software constitutes acceptance of these terms. If you do not agree with these terms, please do not use, install, modify or redistribute this Apple software.

In consideration of your agreement to abide by the following terms, and
subject to these terms, Apple grants you a personal, non-exclusive
license, under Apple's copyrights in this original Apple softwa
```

re (the  
"Apple Software"), to use, reproduce, modify and redistribute t  
he Apple  
Software, with or without modifications, in source and/or binar  
y forms;  
provided that if you redistribute the Apple Software in its ent  
irety and  
without modifications, you must retain this notice and the foll  
owing  
text and disclaimers in all such redistributions of the Apple S  
oftware.  
Neither the name, trademarks, service marks or logos of Apple I  
nc. may  
be used to endorse or promote products derived from the Apple S  
oftware  
without specific prior written permission from Apple. Except a  
s  
expressly stated in this notice, no other rights or licenses, e  
xpress or  
implied, are granted by Apple herein, including but not limited  
to any  
patent rights that may be infringed by your derivative works or  
by other  
works in which the Apple Software may be incorporated.

The Apple Software is provided by Apple on an "AS IS" basis. A  
PPLE  
MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMI  
TATION  
THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND  
FITNESS  
FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS U  
SE AND  
OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, IN  
CIDENTAL  
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUR  
EMENT OF  
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

BUSINESS  
INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,  
MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER  
CAUSED  
AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),  
STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF  
THE  
POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2014 Apple Inc. All Rights Reserved.

\*/

```
#import "UIImageEffects.h"

#import Accelerate;

@implementation UIImageEffects

#pragma mark -
#pragma mark - Effects

//| -----
-----
+ (UIImage *)imageByApplyingLightEffectToImage:(UIImage*)inputImage
{
    UIColor *tintColor = [UIColor colorWithWhite:1.0 alpha:0.3];
    return [self imageByApplyingBlurToImage:inputImage withRadius:60 tintColor:tintColor saturationDeltaFactor:1.8 maskImage:nil];
}

//| -----
-----
+ (UIImage *)imageByApplyingExtraLightEffectToImage:(UIImage*)inputImage
{
```

```
UIColor *tintColor = [UIColor colorWithWhite:0.97 alpha:0.82];
};

return [self imageByApplyingBlurToImage:inputImage withRadius:40 tintColor:tintColor saturationDeltaFactor:1.8 maskImage:nil];
};

//| -----
-----

+ (UIImage *)imageByApplyingDarkEffectToImage:(UIImage*)inputImage
{
    UIColor *tintColor = [UIColor colorWithWhite:0.11 alpha:0.73];
    return [self imageByApplyingBlurToImage:inputImage withRadius:40 tintColor:tintColor saturationDeltaFactor:1.8 maskImage:nil];
};

//| -----
-----

+ (UIImage *)imageByApplyingTintEffectWithColor:(UIColor *)tintColor toImage:(UIImage*)inputImage
{
    const CGFloat EffectColorAlpha = 0.6;
    UIColor *effectColor = tintColor;
    size_t componentCount = CGColorGetNumberOfComponents(tintColor.CGColor);
    if (componentCount == 2) {
        CGFloat b;
        if ([tintColor getWhite:&b alpha:NULL]) {
            effectColor = [UIColor colorWithWhite:b alpha:EffectColorAlpha];
        }
    }
    else {
        CGFloat r, g, b;
        if ([tintColor getRed:&r green:&g blue:&b alpha:NULL]) {
```

```

        effectColor = [UIColor colorWithRed:r green:g blue:b
alpha:EffectColorAlpha];
    }
}

return [self imageByApplyingBlurToImage:inputImage withRadius:20 tintColor:effectColor saturationDeltaFactor:-1.0 maskImage:nil];
}

#pragma mark -
#pragma mark - Implementation

// | -----
-----

+ (UIImage*)imageByApplyingBlurToImage:(UIImage*)inputImage withRadius:(CGFloat)blurRadius tintColor:(UIColor *)tintColor saturationDeltaFactor:(CGFloat)saturationDeltaFactor maskImage:(UIImage *)maskImage
{
#define ENABLE_BLUR 1
#define ENABLE_SATURATION_ADJUSTMENT 1
#define ENABLE_TINT 1

// Check pre-conditions.
if (inputImage.size.width < 1 || inputImage.size.height < 1)
{
    NSLog(@"*** error: invalid size: (%.2f x %.2f). Both dimensions must be >= 1: %@", inputImage.size.width, inputImage.size.height, inputImage);
    return nil;
}
if (!inputImage.CGImage)
{
    NSLog(@"*** error: inputImage must be backed by a CGImage: %@", inputImage);
    return nil;
}
if (maskImage && !maskImage.CGImage)
{
    NSLog(@"*** error: effectMaskImage must be backed by a C

```

```
    CGImage: %@", maskImage);
    return nil;
}

BOOL hasBlur = blurRadius > __FLT_EPSILON__;
BOOL hasSaturationChange = fabs(saturationDeltaFactor - 1.) > __FLT_EPSILON__;

CGImageRef inputCGImage = inputImage.CGImage;
CGFloat inputImageScale = inputImage.scale;
CGBitmapInfo inputImageBitmapInfo = CGImageGetBitmapInfo(inputCGImage);
CGImageAlphaInfo inputImageAlphaInfo = (inputImageBitmapInfo & kCGBitmapAlphaInfoMask);

CGSize outputImageSizeInPoints = inputImage.size;
CGRect outputImageRectInPoints = { CGPointMakeZero, outputImageSizeInPoints };

// Set up output context.
BOOL useOpaqueContext;
if (inputImageAlphaInfo == kCGImageAlphaNone || inputImageAlphaInfo == kCGImageAlphaNoneSkipLast || inputImageAlphaInfo == kCGImageAlphaNoneSkipFirst)
    useOpaqueContext = YES;
else
    useOpaqueContext = NO;
UIGraphicsBeginImageContextWithOptions(outputImageRectInPoints.size, useOpaqueContext, inputImageScale);
CGContextRef outputContext = UIGraphicsGetCurrentContext();
CGContextScaleCTM(outputContext, 1.0, -1.0);
CGContextTranslateCTM(outputContext, 0, -outputImageRectInPoints.size.height);

if (hasBlur || hasSaturationChange)
{
    vImage_Buffer effectInBuffer;
    vImage_Buffer scratchBuffer1;

    vImage_Buffer *inputBuffer;
```

```

    vImage_Buffer *outputBuffer;

    vImage_CGImageFormat format = {
        .bitsPerComponent = 8,
        .bitsPerPixel = 32,
        .colorSpace = NULL,
        // (kCGImageAlphaPremultipliedFirst | kCGBitmapByteOrder32Little)
        // requests a BGRA buffer.
        .bitmapInfo = kCGImageAlphaPremultipliedFirst | kCGBitmapByteOrder32Little,
        .version = 0,
        .decode = NULL,
        .renderingIntent = kCGColorRenderingIntentDefault
    };

    vImage_Error e = vImageBuffer_InitWithCGImage(&effectInBuffer, &format, NULL, inputImage.CGImage, kvImagePrintDiagnosticstoConsole);
    if (e != kvImageNoError)
    {
        NSLog(@"*** error: vImageBuffer_InitWithCGImage returned error code %zi for inputImage: %@", e, inputImage);
        UIGraphicsEndImageContext();
        return nil;
    }

    vImageBuffer_Init(&scratchBuffer1, effectInBuffer.height, effectInBuffer.width, format.bitsPerPixel, kvImageNoFlags);
    inputBuffer = &effectInBuffer;
    outputBuffer = &scratchBuffer1;

#if ENABLE_BLUR
    if (hasBlur)
    {
        // A description of how to compute the box kernel width from the Gaussian
        // radius (aka standard deviation) appears in the SVG spec:
        // http://www.w3.org/TR/SVG/filters.html#feGaussianB

```

```

lurElement
    //
    // For larger values of 's' (s >= 2.0), an approximation can be used: Three
        // successive box-blurs build a piece-wise quadratic convolution kernel, which
            // approximates the Gaussian kernel to within roughly 3%.
    //
    // let d = floor(s * 3*sqrt(2*pi)/4 + 0.5)
    //
    // ... if d is odd, use three box-blurs of size 'd', centered on the output pixel.
    //
    CGFloat inputRadius = blurRadius * inputImageScale;
    if (inputRadius - 2. < __FLT_EPSILON__)
        inputRadius = 2.;
    uint32_t radius = floor((inputRadius * 3. * sqrt(2 * M_PI) / 4 + 0.5) / 2);

    radius |= 1; // force radius to be odd so that the three box-blur methodology works.

    NSInteger tempBufferSize = vImageBoxConvolve_ARGB8888
8(inputBuffer, outputBuffer, NULL, 0, 0, radius, radius, NULL, k
vImageGetTempBufferSize | kvImageEdgeExtend);
    void *tempBuffer = malloc(tempBufferSize);

    vImageBoxConvolve_ARGB8888(inputBuffer, outputBuffer,
, tempBuffer, 0, 0, radius, radius, NULL, kvImageEdgeExtend);
    vImageBoxConvolve_ARGB8888(outputBuffer, inputBuffer,
, tempBuffer, 0, 0, radius, radius, NULL, kvImageEdgeExtend);
    vImageBoxConvolve_ARGB8888(inputBuffer, outputBuffer,
, tempBuffer, 0, 0, radius, radius, NULL, kvImageEdgeExtend);

    free(tempBuffer);

    vImage_Buffer *temp = inputBuffer;
    inputBuffer = outputBuffer;
    outputBuffer = temp;

```

```

    }

#endif

#if ENABLE_SATURATION_ADJUSTMENT
    if (hasSaturationChange)
    {
        CGFloat s = saturationDeltaFactor;
        // These values appear in the W3C Filter Effects spec
        // https://dvcs.w3.org/hg/FXTF/raw-file/default/filters/index.html#grayscaleEquivalent
        // floatingPointSaturationMatrix[] = {
        //     0.0722 + 0.9278 * s,  0.0722 - 0.0722 * s,  0.07
        //     22 - 0.0722 * s,  0,
        //     0.7152 - 0.7152 * s,  0.7152 + 0.2848 * s,  0.71
        //     52 - 0.7152 * s,  0,
        //     0.2126 - 0.2126 * s,  0.2126 - 0.2126 * s,  0.21
        //     26 + 0.7873 * s,  0,
        //     0,                  0,                  0,
        //     1,
        // };
        const int32_t divisor = 256;
        NSUInteger matrixSize = sizeof(floatingPointSaturationMatrix)/sizeof(floatingPointSaturationMatrix[0]);
        int16_t saturationMatrix[matrixSize];
        for (NSUInteger i = 0; i < matrixSize; ++i) {
            saturationMatrix[i] = (int16_t)roundf(floatingPointSaturationMatrix[i] * divisor);
        }
        vImageMatrixMultiply_ARGB8888(inputBuffer, outputBuffer, saturationMatrix, divisor, NULL, NULL, kvImageNoFlags);

        vImage_Buffer *temp = inputBuffer;
        inputBuffer = outputBuffer;
        outputBuffer = temp;
    }
#endif

CGImageRef effectCGImage;

```

```

        if ( (effectCGImage = vImageCreateCGImageFromBuffer(inputBuffer, &format, &cleanupBuffer, NULL, kvImageNoAllocate, NULL) == NULL ) {
            effectCGImage = vImageCreateCGImageFromBuffer(inputBuffer, &format, NULL, NULL, kvImageNoFlags, NULL);
            free(inputBuffer->data);
        }
        if (maskImage) {
            // Only need to draw the base image if the effect image will be masked.
            CGContextDrawImage(outputContext, outputImageRectInPoints, inputCGImage);
        }

        // draw effect image
        CGContextSaveGState(outputContext);
        if (maskImage)
            CGContextClipToMask(outputContext, outputImageRectInPoints, maskImage.CGImage);
        CGContextDrawImage(outputContext, outputImageRectInPoints, effectCGImage);
        CGContextRestoreGState(outputContext);

        // Cleanup
        CGImageRelease(effectCGImage);
        free(outputBuffer->data);
    }
    else
    {
        // draw base image
        CGContextDrawImage(outputContext, outputImageRectInPoints, inputCGImage);
    }

#endif ENABLE_TINT
// Add in color tint.
if (tintColor)
{
    CGContextSaveGState(outputContext);
    CGContextSetFillColorWithColor(outputContext, tintColor);
}

```

```

CGColor);
    CGContextFillRect(outputContext, outputImageRectInPoints);
    CGContextRestoreGState(outputContext);
}
#endif

// Output image is ready.
UIImage *outputImage = UIGraphicsGetImageFromCurrentImageContext();
UIGraphicsEndImageContext();

return outputImage;
#endif ENABLE_BLUR
#endif ENABLE_SATURATION_ADJUSTMENT
#endif ENABLE_TINT
}

// -----
// Helper function to handle deferred cleanup of a buffer.
//
void cleanupBuffer(void *userData, void *buf_data)
{ free(buf_data); }

@end

```

使用

```

- (IBAction)updateImage:(id)sender
{
    NSString *effectText = @"";
    UIImage *effectImage = nil;

    switch (self.imageIndex)
    {
        case 0:
            effectImage = self.image;

```

```
        break;
    case 1:
        effectImage = [UIImageEffects imageByApplyingLightEffectToImage:self.image];
        effectText = NSLocalizedString(@"Light", @"");
        self.effectLabel.textColor = [UIColor darkTextColor]
    ;
        break;
    case 2:
        effectImage = [UIImageEffects imageByApplyingExtraLightEffectToImage:self.image];
        effectText = NSLocalizedString(@"Extra light", @"");
        self.effectLabel.textColor = [UIColor darkTextColor]
    ;
        break;
    case 3:
        effectImage = [UIImageEffects imageByApplyingDarkEffectToImage:self.image];
        effectText = NSLocalizedString(@"Dark", @"");
        self.effectLabel.textColor = [UIColor lightTextColor]
    ;
        break;
    case 4:
        effectImage = [UIImageEffects imageByApplyingTintEffectWithColor:[UIColor blueColor] toImage:self.image];
        effectText = NSLocalizedString(@"Color tint", @"");
        self.effectLabel.textColor = [UIColor lightTextColor]
    ;
        break;
    }

    self.imageView.image = effectImage;
    self.effectLabel.text = effectText;

}
```

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-04-06 | Alfred Jiang | -  |

## 方案名称

UILabel - UILabel 边缘黑线问题

## 关键字

UILabel \ 边缘黑线

## 需求场景

1. 代码实现 UILabel 时出现的边缘黑线问题

## 参考链接

1. 纯色UILabel右边缘的黑线问题
2. UILabel右边缘黑线问题

## 详细内容

当 UILabel 的宽高出现浮点型数字时，在高清屏下有可能出现右侧边缘黑线问题，解决方法是使用 `CGRectIntegral()` 方法确保 frame 中不存在非整形数值

```
label.frame = CGRectMakeIntegral(CGRectMake(index_x, 0, ceil(width),  
40));
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-27 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UILabel - 悬浮美金单位效果的实现

## 关键字

UILabel \ 悬浮 \ 单位 \ 美金单位

## 需求场景

1. 需要实现如效果图显示的特殊的美金单位显示效果时

## 参考链接

(无)

## 详细内容

1. 在  $Xib$  条件下，用于显示金额数字的 *UILabelA* 的 *Autolayout* 约束条件不要设置长度，这样 *UILabelA* 可以根据内容自动设置自己的长度；
2. 设置显示 \$ 的 *UILabelB* 尾部约束条件为距离 *UILabelA* 顶部  $X$  个像素，这样最终就会显示出如图所示的效果

## 效果图



## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注              |
|----|------------|--------------|-----------------|
| 1  | 2015-04-13 | Alfred Jiang | -               |
| 2  | 2015-08-28 | Alfred Jiang | -               |
| 3  | 2015-12-23 | Alfred Jiang | -               |
| 3  | 2016-05-03 | Alfred Jiang | 增加 批量替换子字符串显示格式 |

## 方案名称

UILabel - 显示多格式文本

## 关键字

UILabel \ NSMutableAttributedString \ attributedText \ 批量替换

## 需求场景

1. 需要显示一段多格式的文本
2. 批量替换子字符串显示格式

## 参考链接

1. [Stack Overflow - Change attributes of substrings in a NSAttributedString](#)

## 详细内容

### 1. Objective-C 示例

显示富文本格式

```
NSString *plainText = [NSString stringWithFormat:@"%@", @"click"];
NSMutableAttributedString *styleText = [[NSMutableAttributedString alloc] initWithString:plainText];
NSDictionary *attr = @{NSFontAttributeName: [UIFont systemFontOfSize:15], NSForegroundColorAttributeName : [UIColor redColor]};
NSRange nameRange = [plainText rangeOfString:aInfo.title];
[styleText setAttributes:attr range:nameRange];
self.labelInfo.attributedText = styleText;
```

批量替换子字符串显示格式

定义：

```
- (NSMutableAttributedString *)setSubString:(NSString*)substring
inString:(NSMutableAttributedString *)mutableAttributedString w
ithColor:(UIColor *)color andFont:(UIFont *)font
{
    NSRegularExpression *regex = [NSRegularExpression regularExp
ressionWithPattern:[NSString stringWithFormat:@"(%@)", substring]
options:kNilOptions error:nil];

    NSString *string = [mutableAttributedString string];

    NSRange range = NSMakeRange(0, string.length);

    [regex enumerateMatchesInString:string options:kNilOptions r
ange:range usingBlock:^(NSTextCheckingResult *result, NSMatching
Flags flags, BOOL *stop) {

        NSRange subStringRange = [result rangeAtIndex:1];
        [mutableAttributedString addAttribute:NSForegroundColorA
ttributeName value:color range:subStringRange];
        [mutableAttributedString addAttribute:NSFontAttributeName
value:font range:subStringRange];
    }];
}

return mutableAttributedString;
}
```

使用：

```
NSMutableAttributedString *mString = [[NSMutableAttributedString alloc] initWithString:@"This is the text and i want to replace something and stuff and stuff"];  
  
mString = [self setSubString:@"stuff" inString:mString withColor:[UIColor greenColor] andFont:[UIFont systemFontOfSize:14.0]];  
mString = [self setSubString:@"and" inString:mString withColor:[UIColor redColor] andFont:[UIFont systemFontOfSize:14.0]];  
  
self.labelInfo.attributedText = mString;
```

效果：

This is the text **and** i want to replace something **and** **stuff** **and** **stuff**

## 2. Swift 示例

示例1

```
var plainText = title! + " button pressed";  
var styleText = NSMutableAttributedString(string: plainText);  
var attributes : NSDictionary = [NSFontAttributeName : UIFont.boldSystemFontOfSize( statusLabel.font.pointSize )]  
let nameRange : NSRange = (plainText as NSString).rangeOfString(title!);  
//styleText.setAttributes(attributes, )  
styleText.setAttributes(attributes, range: nameRange);  
statusLabel.attributedText = styleText;
```

示例2

```
func attributedString(plainText : NSString, sString : NSString)
-> NSMutableAttributedString
{
    var styleText = NSMutableAttributedString(string: plainText)
    var font : UIFont = UIFont(name: "Helvetica-BoldOblique", size: 13.0)!
    var color : UIColor = COLOR_ORANGE
    var attrs = [NSFontAttributeName : font, NSForegroundColorAttributeName : color]
    let nameRange : NSRange = (plainText as NSString).rangeOfString(sString)
    styleText.setAttributes(attrs, range: nameRange)
    return styleText
}
```

## 效果图

(无)

## 备注

- [UILabel \ UITextField - 设置行间距等富文本属性](#)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-31 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UILabel - 显示换行的方法

## 关键字

UILabel \ 换行

## 需求场景

1. 需要对 UILabel 中字符串进行换行显示时

## 参考链接

1. [CocoaChina - UILabel显示换行的方法](#)

## 详细内容

```
UILabel*label;  
  
//设置换行  
label.lineBreakMode = UILineBreakModeWordWrap;  
label.numberOfLines = 0;  
  
//换行符还是\n  
比如NSString * xstring=@"lineone\nlinetwo"
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注           |
|----|------------|--------------|--------------|
| 1  | 2015-05-07 | Alfred Jiang | -            |
| 2  | 2015-08-18 | Alfred Jiang | 更新OC版本iOS6情况 |
| 3  | 2015-12-23 | Alfred Jiang | -            |

## 方案名称

UILabel - 计算文本高度

## 关键字

UILabel \ 高度 \ 文本高度 \ UILabel 高度

## 需求场景

1. 需要根据文本内容动态修改控件高度时

## 参考链接

1. [CSDN - iOS 7下计算文本高度注意事项](#)

## 详细内容

Objective-C 调用示例

```

- (CGSize)string:(NSString *)string rectSize:(CGSize)upperSize font:(UIFont *)aFont
{
    CGSize labelsize = CGSizeMake(0, 0);

    BOOL isIOS7 = ([getOsVersion() floatValue] >= 7.0);
    if (isIOS7) {
        NSDictionary *dic = [NSDictionary dictionaryWithObjectsAndKeys:aFont, NSFontAttributeName, nil];

        labelsize = [string boundingRectWithSize:upperSize
                                         options:\n
                                         NSStringDrawingTruncatesLastVisibleLine
                                         |
                                         NSStringDrawingUsesLineFragmentOrigin |
                                         NSStringDrawingUsesFontLeading
                                         attributes:dic
                                         context:nil].size;
    }
    else
    {
        labelsize = [string sizeWithFont:aFont constrainedToSize:upperSize lineBreakMode:NSLineBreakByWordWrapping];
    }

    return labelsize;
}

```

swift 调用示例（swift最低支持到iOS7，故不考虑iOS6情况）

```

func sizeWithString(aString : NSString) -> CGFloat
{
    let rect : CGSize = REXOCTools.string(aString, rectSize: CGSizeMake(self.labelRolesResponsibilities.frame.width, CGFloat.max), font: UIFont(name: "Helvetica-Light", size: 13.0))
    return ceil(rect.height)
}

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-31 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UILabel - 通过字符串长度计算显示框大小的方法

## 关键字

UILabel \ Size \ UITextView \ CGSize \ CGFrame

## 需求场景

1. 根据一段文字动态计算出用于显示的 UILabel 或 UITextView 等控件的显示大小

## 参考链接

**Swift** 版本

**Objective-C** 版本

```
+ (CGSize)string:(NSString *)string rectSize:(CGSize)upperSize font:(UIFont *)aFont
{
    NSDictionary *dic = [NSDictionary dictionaryWithObjectsAndKeys:aFont, NSFontAttributeName, nil];

    CGSize size = [string boundingRectWithSize:upperSize
                                         options:\n
                                         NSStringDrawingTruncatesLastVisibleLine
                                         |
                                         NSStringDrawingUsesLineFragmentOrigin |
                                         NSStringDrawingUsesFontLeading
                                         attributes:dic
                                         context:nil].size
;

    return size;
}
```

## 详细内容

### 效果图

(无)

### 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注        |
|----|------------|--------------|-----------|
| 1  | 2015-10-26 | Alfred Jiang | -         |
| 2  | 2015-12-21 | Alfred Jiang | -         |
| 2  | 2016-01-25 | Alfred Jiang | 更新富文本属性举例 |

## 方案名称

UILabel \ UITextField - 设置行间距等富文本属性

## 关键字

UILabel \ UITextField \ 间距 \ 行间距 \ 富文本

## 需求场景

1. 需要对 UILabel / UITextField 行间距等富文本属性进行调整时

## 参考链接

1. [IT985 博客 - UILabel UITextField 调整行间距](#)
2. [CSDN - iOS NSMutableAttributedString/NSAttributedString 富文本设置](#)
3. [OSChina - iOS 字符属性NSAttributedString 描述](#)
4. [博客园 - 给iOS开发新手送点福利, 简述文本属性Attributes的用法\(推荐\)](#)

## 详细内容

```

// 设置行间距
- (void)setLineSpacing:(CGFloat)spacing label:(UILabel *)label
{
    NSMutableAttributedString * attributedString = [[NSMutableAttributedString alloc] initWithString:label.text];
    NSMutableParagraphStyle * paragraphStyle = [[NSMutableParagraphStyle alloc] init];
    [paragraphStyle setLineSpacing:spacing];
    [attributedString addAttribute:NSStrongTextAttribute
        value:paragraphStyle range:NSMakeRange(0, [label.text length])];
    [label setAttributedText:attributedString];
    [label sizeToFit];
}

```

### NSMutableParagraphStyle 属性举例

- alignment //对齐方式
- firstLineHeadIndent //首行缩进
- headIndent //缩进
- tailIndent //尾部缩进
- lineBreakMode //断行方式
- maximumLineHeight //最大行高
- minimumLineHeight //最低行高
- lineSpacing //行距
- paragraphSpacing //段距
- paragraphSpacingBefore //段首空间
- baseWritingDirection //句子方向
- lineHeightMultiple //可变行高,乘因数。
- hyphenationFactor //连字符属性

**NSMutableAttributedString** 字符属性（Attribute）举例

- **NSFontAttributeName** // (字体) 该属性所对应的值是一个 **UIFont** 对象。该属性用于改变一段文本的字体。如果不指定该属性，则默认为 12-point Helvetica(Neue)。
- **NSParagraphStyleAttributeName** // (段落) 该属性所对应的值是一个 **NSParagraphStyle** 对象。该属性在一段文本上应用多个属性。如果不指定该属性，则默认为 **NSParagraphStyle** 的 **defaultParagraphStyle** 方法返回的默认段落属性。
- **NSForegroundColorAttributeName** // (字体颜色) 该属性所对应的值是一个 **UIColor** 对象。该属性用于指定一段文本的字体颜色。如果不指定该属性，则默认为黑色。
- **NSBackgroundColorAttributeName** // (字体背景色) 该属性所对应的值是一个 **UIColor** 对象。该属性用于指定一段文本的背景颜色。如果不指定该属性，则默认无背景色。
- **NSLigatureAttributeName** // (连字符) 该属性所对应的值是一个 **NSNumber** 对象(整数)。连体字符是指某些连在一起的字符，它们采用单个的图元符号。0 表示没有连体字符。1 表示使用默认的连体字符。2 表示使用所有连体符号。默认值为 1 (注意，iOS 不支持值为 2)。
- **NSKernAttributeName** // (字间距) 该属性所对应的值是一个 **NSNumber** 对象(整数)。字母紧排指定了用于调整字距的像素点数。字母紧排的效果依赖于字体。值为 0 表示不使用字母紧排。默认值为 0。
- **NSStrikethroughStyleAttributeName** // (删除线) 该属性所对应的值是一个 **NSNumber** 对象(整数)。该值指定是否在文字上加上删除线，该值参考“Underline Style Attributes”。默认值是 **NSUnderlineStyleNone**。

```

enum {
    NSUnderlineStyleNone = 0x00,
    NSUnderlineStyleSingle = 0x01,
    ...
};

// 设置删除线。

```

- **NSUnderlineStyleAttributeName** // (下划线) 该属性所对应的值是一个 **NSNumber** 对象(整数)。该值指定是否在文字上加上下划线，该值参考“Underline Style Attributes”。默认值是**NSUnderlineStyleNone**。
- **NSStrokeColorAttributeName** // (边线颜色) 该属性所对应的值是一个 **UIColor** 对象。如果该属性不指定(默认)，则等同于 **NSForegroundColorAttributeName**。否则，指定为删除线或下划线颜色。更多细节见“Drawing attributedstrings that are both filled and stroked”。
- **NSStrokeWidthAttributeName** // (边线宽度) 该属性所对应的值是一个 **NSNumber** 对象(小数)，。该值改变描边宽度(相对于字体**size**的百分比)(设置笔画的粗细)。默认为 0，即不改变。正数只改变描边宽度。负数同时改变文字的描边和填充宽度。例如，对于常见的空心字，这个值通常为 3.0。
- **NSShadowAttributeName** // (阴影) 该属性所对应的值是一个 **NSShadow** 对象。设置笔画的阴影，默认为 **nil**。
- **NSVerticalGlyphFormAttributeName** // (横竖排版) 该属性所对应的值是一个 **NSNumber** 对象(整数)。0 表示横排文本。1 表示竖排文本。在 iOS 中，总是使用横排文本，0 以外的值都未定义。

## 效果图

(无)

## 备注

- [UILabel - 显示多格式文本](#)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注      |
|----|------------|--------------|---------|
| 1  | 2015-03-23 | Alfred Jiang | -       |
| 2  | 2015-03-31 | Alfred Jiang | 更新注意问题1 |
| 3  | 2015-12-23 | Alfred Jiang | -       |

## 方案名称

UINavigationBar - 自定义按钮和标题

## 关键字

UINavigationBar \ navigationItem \ titleView

## 需求场景

1. 需要自定义 UINavigationBar 的按钮和标题等

## 参考链接

1. 博客园 - 学习笔记 : [UINavigationbar的背景修改方法集合](#)
2. CSDN - [自定义UINavigationBar的属性](#)

## 详细内容

### Swift 版本

#### 1. 通过 **xib** 文件自定义 **navigationItem** 的 **titleView**

1. 实例化自定义对象

```
var menuTitle : REXMenuBtnView = REXMenuBtnView.loadFromNibNamed("REXMenuBtnView", bundle: nil) as REXMenuBtnView
```

## 2. 在 `viewDidLoad` 中初始化必要的属性

```
menuTitle.btnAddTarget(self, action: Selector("clickTitleBtn:"), forControlEvents: UIControlEvents.TouchUpInside)
    //示例添加按钮响应
menuTitle.setSelected(false)      //示例调用方法
```

## 3. 在 `viewDidLayoutSubviews` 中赋值 `navigationItem`

```
self.navigationItem.titleView = menuTitle
```

## 4. Swift 下载入自定义 `xib` View 方法

```
extension UIView {
    class func loadFromNibNamed(nibNamed: String, bundle : NSBundle? = nil) -> UIView? {
        return UINib(nibName: nibNamed, bundle: bundle).instantiateWithOwner(nil, options: nil)[0] as? UIView
    }
}
```

## 2. 通过 `UILabel` 修改 `navigationItem` 的 `TitleView`

```
var labelTitle : UILabel = UILabel(frame: CGRectMake(0, 0, 100, 44))
labelTitle.font = UIFont(name: "Georgia-Italic", size: 18.0)
labelTitle.textColor = UIColor(red: 73.0 / 255.0, green: 73.0 / 255.0, blue: 73.0 / 255.0, alpha: 1.0)
labelTitle.backgroundColor = UIColor.clearColor()
labelTitle.textAlignment = NSTextAlignment.Center
labelTitle.text = "Message"
self.navigationItem.titleView = labelTitle
```

## 3. 自定义 `NavigationBar` 背景图

```
self.navigationController?.navigationBar.setBackgroundImage(UIImage(named: "login_Bg@2x"), forBarMetrics: UIBarMetrics.Default)
```

#### 4. 自定义 **NavigationBar** 按钮

```
var backBtn : MKButton = REXTools.navRoundBtn("", frame : CGRectMake(0, 0, 12, 20), color: COLOR_GRAY,imageName: "arrowLeft", target: self, action: Selector("clickBackBtn:"))
var tempItem : UIBarButtonItem = UIBarButtonItem(customView: backBtn)
self.navigationItem.leftBarButtonItems = [tempItem]
```

### Objective-C 版本

```

//NavigationBar设置背景图
[_navigationBar_ setBackgroundImage:[UIImage imageNamed:@"__导航条背景图__"] forBarMetrics:UIBarMetricsDefault];

//LeftButton设置属性
UIButton *leftButton = [UIButton buttonWithType:UIButtonTypeCustom];
[leftButton setFrame:CGRectMake(0, 0, __leftButtonWidth__, __leftButtonHeight__)];
[leftButton setBackgroundImage:[UIImage imageNamed:@"__左按钮背景图__"] forState:UIControlStateNormal];
[leftButton addTarget:self action:@selector(__leftButtonSelector__)
forControlEvents:UIControlEventTouchUpInside];
[_leftBarButton_ setCustomView:leftButton];

//RightButton设置属性
UIButton *rightButton = [UIButton buttonWithType:UIButtonTypeCustom];
[rightButton setFrame:CGRectMake(0, 0, __rightButtonWidth__, __rightButtonHeight__)];
[rightButton setBackgroundImage:[UIImage imageNamed:@"__右按钮背景图__"] forState:UIControlStateNormal];
[rightButton addTarget:self action:@selector(__rightBarButton__)
forControlEvents:UIControlEventTouchUpInside];
[_rightBarButton_ setCustomView:rightButton];

//NavigationItem设置属性
UILabel *titleLabel = [[UILabel alloc] initWithFrame:CGRectMake(0
, 0, 320, 44)];
titleLabel.font = [UIFont boldSystemFontOfSize:_fontOfSize__];
titleLabel.textColor = __textColor__;
titleLabel.backgroundColor = [UIColor clearColor];
titleLabel.textAlignment = UITextAlignmentCenter;
titleLabel.text = @"__text__";
_navigationItem_.titleView = titleLabel;

```

## 注意问题

## 1. 设置 **translucent** 属性避免 **UIView** 向上偏移

```
UINavigationBar.appearance().translucent = false
```

以上写法在 *iOS 7* 环境下易造成 *Crash*，需要加如下条件

```
if (UIDevice.currentDevice().systemVersion as NSString).floatValue >= 8.0 {  
    UINavigationBar.appearance().translucent = false  
}
```

或者使用下面的写法设置避免 **UIView** 向上偏移

```
self.navigationController?.navigationBar.translucent = false
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-22 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UINavigationController - 使用 SlideNavigationController 实现侧滑需求

## 关键字

UINavigationController \ 侧滑显示

## 需求场景

1. 需要实现侧滑需求时

## 参考链接

1. [GitHub - SlideNavigationController](#)

## 详细内容

### 1. SlideNavigationController.h

```
//  
//  SlideNavigationController.h  
//  SlideMenu  
//  
//  Created by Aryan Gh on 4/24/13.  
//  Copyright (c) 2013 Aryan Ghassemi. All rights reserved.  
//  
//  https://github.com/aryaxt/iOS-Slide-Menu  
//
```

```

// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to deal
// in the Software without restriction, including without limitation the rights
// to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
// EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
// OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
// ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
// THE SOFTWARE.

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>

@protocol SlideNavigationControllerDelegate <NSObject>
@optional
- (BOOL)slideNavigationControllerShouldDisplayRightMenu;
- (BOOL)slideNavigationControllerShouldDisplayLeftMenu;
@end

typedef enum{

```

```

    MenuLeft,
    MenuRight,
}Menu;

@interface SlideNavigationController : UINavigationController <U
INavigationControllerDelegate>

@property (nonatomic, assign) BOOL avoidSwitchingToSameClassView
Controller;
@property (nonatomic, assign) BOOL enableSwipeGesture;
@property (nonatomic, strong) UIViewController *rightMenu;
@property (nonatomic, strong) UIViewController *leftMenu;
@property (nonatomic, strong) UIBarButtonItem *leftBarButtonItem
;
@property (nonatomic, strong) UIBarButtonItem *rightBarButtonItem
m;

+ (SlideNavigationController *)sharedInstance;
- (void)switchToViewController:(UIViewController *)viewController
withCompletion:(void (^)())completion;

@end

```

## 1. SlideNavigationController.m

```

//
//  SlideNavigationController.m
//  SlideMenu
//
//  Created by Aryan Gh on 4/24/13.
//  Copyright (c) 2013 Aryan Ghassemi. All rights reserved.
//
//  https://github.com/aryaxt/iOS-Slide-Menu
//
// Permission is hereby granted, free of charge, to any person o
btaining a copy
// of this software and associated documentation files (the "Sof
tware"),
// to deal
// in the Software without restriction, including without limita

```

```
tion the rights
// to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell
// copies of the Software, and to permit persons to whom the Sof
ftware is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall b
e included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIN
D, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERC
HANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO E
VENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE
E, ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN
// THE SOFTWARE.

#import "SlideNavigationController.h"

@interface SlideNavigationController()
@property (nonatomic, strong) UITapGestureRecognizer *tapRecogni
zer;
@property (nonatomic, strong) UIPanGestureRecognizer *panRecogni
zer;
@property (nonatomic, assign) CGPoint draggingPoint;
@end

@implementation SlideNavigationController
@synthesize rightMenu;
@synthesize leftMenu;
@synthesize tapRecognizer;
@synthesize panRecognizer;
```

```
@synthesize draggingPoint;
@synthesize leftBarButtonItem;
@synthesize rightBarButtonItem;
@synthesize enableSwipeGesture;

#define MENU_OFFSET 60
#define MENU_SLIDE_ANIMATION_DURATION .3
#define MENU_QUICK_SLIDE_ANIMATION_DURATION .1
#define MENU_IMAGE @"Icon-Menu"//@"menu-button"

static SlideNavigationController *singletonInstance;

#pragma mark - Initialization -

+ (SlideNavigationController *)sharedInstance
{
    return singletonInstance;
}

- (void)awakeFromNib
{
    [self setup];
}

- (id)initWithRootViewController:(UIViewController *)rootViewController
{
    if (self = [super initWithRootViewController:rootViewController])
    {
        [self setup];
    }

    return self;
}

- (id)init
{
    if (self = [super init])
    {
```

```

        [self setup];
    }

    return self;
}

- (void)setup
{
    self.avoidSwitchingToSameClassViewController = YES;
    singletonInstance = self;
    self.delegate = self;

    self.view.layer.shadowColor = [UIColor darkGrayColor].CGColor;
    self.view.layer.shadowRadius = 10;
    self.view.layer.shadowPath = [UIBezierPath bezierPathWithRect:self.view.bounds].CGPath;
    self.view.layer.shadowOpacity = 1;
    self.view.layer.shouldRasterize = YES;
    self.view.layer.rasterizationScale = [UIScreen mainScreen].scale;

    [self setEnableSwipeGesture:YES];
}

#pragma mark - Public Methods -

- (void)switchToViewController:(UIViewController *)viewController
withCompletion:(void (^)(()))completion
{
    if (self.avoidSwitchingToSameClassViewController && [self.to
pViewController isKindOfClass:viewController.class])
    {
        [self closeMenuWithCompletion:completion];
        return;
    }

    __block CGRect rect = self.view.frame;

    if ([self isMenuOpen])

```

```
{  
    [UIView animateWithDuration:MENU_SLIDE_ANIMATION_DURATION  
        delay:0  
        options:UIViewAnimationOptionCurveEaseOut  
        animations:^{
            rect.origin.x = (rect.origin.x > 0) ? rect.size.width : -1*rect.size.width;  
            self.view.frame = rect;  
    } completion:^(BOOL finished) {  
  
        [super popToRootViewControllerAnimated:NO];  
        [super pushViewController:viewController animated:NO];  
    };  
  
    [self closeMenuWithCompletion:^{  
        if (completion)  
            completion();  
    }];  
}];  
}  
else  
{  
    [super popToRootViewControllerAnimated:NO];  
    [super pushViewController:viewController animated:NO];  
  
    if (completion)  
        completion();  
}  
}  
}  
  
- (NSArray *)popToRootViewControllerAnimated:(BOOL)animated  
{  
    if ([self isMenuOpen])  
    {  
        [self closeMenuWithCompletion:^{  
            [super popToRootViewControllerAnimated:animated];  
        }];  
    }  
}
```

```
else
{
    return [super popToRootViewControllerAnimated:animated];
}

return nil;
}

- (void)pushViewController:(UIViewController *)viewController animated:(BOOL)animated
{
    if ([self isMenuOpen])
    {
        [self closeMenuWithCompletion:^{
            [super pushViewController:viewController animated:animated];
        }];
    }
    else
    {
        [super pushViewController:viewController animated:animated];
    }
}

- (NSArray *)popToViewController:(UIViewController *)viewController animated:(BOOL)animated
{
    if ([self isMenuOpen])
    {
        [self closeMenuWithCompletion:^{
            [super popToViewController:viewController animated:animated];
        }];
    }
    else
    {
        return [super popToViewController:viewController animated:animated];
    }
}
```

```
    return nil;
}

#pragma mark - Private Methods -

- (UIBarButtonItem *)barButtonItemForMenu:(Menu)menu
{
    SEL selector = (menu == MenuLeft) ? @selector(leftMenuSelected:) : @selector(rightMenuSelected:);
    UIBarButtonItem *customButton = (menu == MenuLeft) ? self.leftBarButtonItem : self.rightBarButtonItem;

    if (customButton)
    {
        customButton.action = selector;
        customButton.target = self;
        return customButton;
    }
    else
    {
        UIImage *image = [UIImage imageNamed:MENU_IMAGE];
        return [[UIBarButtonItem alloc] initWithImage:image style:UIBarButtonItemStylePlain target:self action:selector];
    }
}

- (BOOL)isMenuOpen
{
    return (self.view.frame.origin.x == 0) ? NO : YES;
}

- (BOOL)shouldDisplayMenu:(Menu)menu forViewController:(UIViewController *)vc
{
    if (menu == MenuRight)
    {
        if ([vc respondsToSelector:@selector(slideNavigationControllerShouldDisplayRightMenu)] &&
            [(UIViewController<SlideNavigationControllerDelegate
```

```

> *)vc slideNavigationControllerShouldDisplayRightMenu])
{
    return YES;
}
if (menu == MenuLeft)
{
    if ([vc respondsToSelector:@selector(slideNavigationControllerShouldDisplayLeftMenu)] &&
        [(UIViewControllerAnimated<SlideNavigationControllerDelegate
> *)vc slideNavigationControllerShouldDisplayLeftMenu])
    {
        return YES;
    }
}

return NO;
}

- (void)openMenu:(Menu)menu withDuration:(float)duration andCompletion:(void (^)())completion
{
    [self.topViewController.view addGestureRecognizer:self.tapRecognizer];

    if (menu == MenuLeft)
    {
        [self.righMenu.view removeFromSuperview];
        [self.view.window insertSubview:self.leftMenu.view atIndex:0];
    }
    else
    {
        [self.leftMenu.view removeFromSuperview];
        [self.view.window insertSubview:self.righMenu.view atIndex:0];
    }

    [UIView animateWithDuration:duration
                          delay:0

```

```

        options:UIViewAnimationOptionCurveEaseOut

    animations:^{
        CGRect rect = self.view.frame;
        rect.origin.x = (menu == MenuLeft) ? (rect.size.width - MENU_OFFSET) : ((rect.size.width - MENU_OFFSET) * -1);
        self.view.frame = rect;
    }
    completion:^(BOOL finished) {
        if (completion)
            completion();
    }];
}

- (void)openMenu:(Menu)menu withCompletion:(void (^)(()))completion
{
    [self openMenu:menu withDuration:MENU_SLIDE_ANIMATION_DURATION andCompletion:completion];
}

- (void)closeMenuWithDuration:(float)duration andCompletion:(void (^)(()))completion
{
    [self.topViewController.view removeGestureRecognizer:self.tapRecognizer];

    [UIView animateWithDuration:duration
                          delay:0
                        options:UIViewAnimationOptionCurveEaseOut

    animations:^{
        CGRect rect = self.view.frame;
        rect.origin.x = 0;
        self.view.frame = rect;
    }
    completion:^(BOOL finished) {
        if (completion)
            completion();
    }];
}

```

```
        }];
    }

    - (void)closeMenuWithCompletion:(void (^)(()))completion
    {
        [self closeMenuWithDuration:MENU_SLIDE_ANIMATION_DURATION
            andCompletion:completion];
    }

#pragma mark - UINavigationControllerDelegate Methods -

    - (void)navigationController:(UINavigationController *)navigationController
        willShowViewController:(UIViewController *)viewController
        animated:(BOOL)animated
    {
        if ([self shouldDisplayMenu:MenuLeft forViewController:viewController])
            viewController.navigationItem.leftBarButtonItem = [self
                barButtonItemForMenu:MenuLeft];

        if ([self shouldDisplayMenu:MenuRight forViewController:viewController])
            viewController.navigationItem.rightBarButtonItem = [self
                barButtonItemForMenu:MenuRight];
    }

#pragma mark - IBActions -

    - (void)leftMenuSelected:(id)sender
    {
        if ([self isMenuOpen])
            [self closeMenuWithCompletion:nil];
        else
            [self openMenu:MenuLeft withCompletion:nil];
    }

    - (void)righttMenuSelected:(id)sender
    {
```

```
if ([self isMenuOpen])
    [self closeMenuWithCompletion:nil];
else
    [self openMenu:MenuRight withCompletion:nil];
}

#pragma mark - Gesture Recognizing -

- (void)tapDetected:(UITapGestureRecognizer *)tapRecognizer
{
    [self closeMenuWithCompletion:nil];
}

- (void)panDetected:(UIPanGestureRecognizer *)aPanRecognizer
{
    static NSInteger velocityForFollowingDirection = 1000;

    CGPoint translation = [aPanRecognizer translationInView:aPanRecognizer.view];
    CGPoint velocity = [aPanRecognizer velocityInView:aPanRecognizer.view];

    if (aPanRecognizer.state == UIGestureRecognizerStateBegan)
    {
        self.draggingPoint = translation;
    }
    else if (aPanRecognizer.state == UIGestureRecognizerStateChanged)
    {
        NSInteger movement = translation.x - self.draggingPoint.x;
        CGRect rect = self.view.frame;
        rect.origin.x += movement;

        if (rect.origin.x >= self minXForDragging && rect.origin.x <= self maxXForDragging)
            self.view.frame = rect;

        self.draggingPoint = translation;
    }
}
```

```

        if (rect.origin.x > 0)
    {
        [self.rightMenu.view removeFromSuperview];
        [self.view.window insertSubview:self.leftMenu.view atIndex:0];
    }
    else
    {
        [self.leftMenu.view removeFromSuperview];
        [self.view.window insertSubview:self.rightMenu.view atIndex:0];
    }
}

else if (aPanRecognizer.state == UIGestureRecognizerStateEnded)
{
    NSInteger currentX = self.view.frame.origin.x;
    NSInteger currentXOffset = (currentX > 0) ? currentX : currentX * -1;
    NSInteger positiveVelocity = (velocity.x > 0) ? velocity.x : velocity.x * -1;

    // If the speed is high enough follow direction
    if (positiveVelocity >= velocityForFollowingDirection)
    {
        // Moving Right
        if (velocity.x > 0)
        {
            if (currentX > 0)
            {
                [self openMenu:(velocity.x > 0) ? MenuLeft : MenuRight withCompletion:nil];
            }
            else
            {
                [self closeMenuWithDuration:MENU_QUICK_SLIDE_ANIMATION_DURATION andCompletion:nil];
            }
        }
        // Moving Left
    }
}

```

```

        else
    {
        if (currentX > 0)
        {
            [self closeMenuWithCompletion:nil];
        }
        else
        {
            Menu menu = (velocity.x > 0) ? MenuLeft : Me
nuRight;

            if ([self shouldDisplayMenu:menu forViewController:self.visibleViewController])
                [self openMenu:(velocity.x > 0) ? MenuLe
ft : MenuRight withDuration:MENU_QUICK_SLIDE_ANIMATION_DURATION
andCompletion:nil];
        }
    }
    else
    {
        if (currentXOffset < self.view.frame.size.width/2)
            [self closeMenuWithCompletion:nil];
        else
            [self openMenu:(currentX > 0) ? MenuLeft : MenuR
ight withCompletion:nil];
    }
}

- (NSInteger)minXForDragging
{
    if ([self shouldDisplayMenu:MenuRight forViewController:self
.topViewController])
    {
        return (self.view.frame.size.width - MENU_OFFSET) * -1;
    }

    return 0;
}

```

```
- (NSInteger)maxXForDragging
{
    if ([self shouldDisplayMenu:MenuLeft forViewController:self.topViewController])
    {
        return self.view.frame.size.width - MENU_OFFSET;
    }

    return 0;
}

#pragma mark - Setter & Getter -

- (UITapGestureRecognizer *)tapRecognizer
{
    if (!tapRecognizer)
    {
        tapRecognizer = [[UITapGestureRecognizer alloc] initWithTarget:self action:@selector(tapDetected:)];
    }

    return tapRecognizer;
}

- (UIPanGestureRecognizer *)panRecognizer
{
    if (!panRecognizer)
    {
        panRecognizer = [[UIPanGestureRecognizer alloc] initWithTarget:self action:@selector(panDetected:)];
    }

    return panRecognizer;
}

- (void)setEnableSwipeGesture:(BOOL)markEnableSwipeGesture
{
    enableSwipeGesture = markEnableSwipeGesture;
```

```
if (enableSwipeGesture)
{
    [self.view addGestureRecognizer:self.panRecognizer];
}
else
{
    [self.view removeGestureRecognizer:self.panRecognizer];
}
@end
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-30 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UIPageControl - 翻页显示的实现

## 关键字

UIPageControl \ 翻页 \ UIScrollView

## 需求场景

1. 需要实现 UIScrollView 翻页显示并显示页码标签时

## 参考链接

(无)

## 详细内容

### 1. Swift 版本

```
func scrollViewDidScroll(scrollView: UIScrollView) {
    self.resetPageControl()
}

func resetPageControl() {
    var iPages : NSInteger = NSInteger(self.scrollViewAttachment
        .contentSize.width / self.scrollViewAttachments.frame.size.width)
    var iPage : NSInteger = NSInteger(self.scrollViewAttachments
        .contentOffset.x / self.scrollViewAttachments.frame.size.width)
    self.pageControlItem.numberOfPages = iPages
    self.pageControlItem.currentPage = iPage
}
```

## 2. Objective-C 版本

```
- (void)scrollViewDidScroll:(UIScrollView *)scrollView
{
    [self resetPageControl];
}

-(void)resetPageControl
{
    NSInteger iPages = self.scrollViewMain.contentSize.width / self.scrollViewMain.frame.size.width;
    NSInteger iPage = self.scrollViewMain.contentOffset.x / self.scrollViewMain.frame.size.width;
    self.pageControlItem.numberOfPages = iPages;
    self.pageControlItem.currentPage = iPage;
}
```

## 3. UIScrollView 添加页面显示元素实现示例

```
func setUpAttachments(attachmens : NSArray)
{
    let pages = CGFloat(ceil(Double(CGFloat(attachmens.count) /
4.0))) //向上取整计算页总数
    self.scrollViewAttachments.contentSize = CGSizeMake(CGFloat((
CGFloat(WIDTHSCREEN - 10.0)) * pages), 100.0)

    for obj in attachmens
    {
        var aAttachment : AuctionAttachment = obj as AuctionAttachment
        var aItemView : REXAttachmentView = REXAttachmentView.lo
adFromNibNamed("REXAttachmentView", bundle: nil) as REXAttachmentView
        aItemView.aAttachment = aAttachment
        self.scrollViewAttachments.addSubview(aItemView)
        aItemView.frame = CGRectMake((WIDTHSCREEN - 10.0) / 4.0
* CGFloat(attachmens.indexOfObject(aAttachment)), 0, (WIDTHSCREEN
- 10.0) / 4.0, 100.0)
        aItemView.setUpView(aAttachment)
        aItemView.delegate = self
    }

    resetPageControl()
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-04-06 | Alfred Jiang | -  |

## 方案名称

UIScrollView - UIScrollView 添加 UIButton 时不响应滑动操作

## 关键字

手势 \ UILabel \ UIView \ 单击手势 \ UIScrollView \ UITableViewCell \ UIButton \ 滑动

## 需求场景

1. 解决 UIScrollView \ UITableViewCell 添加 UIButton 时不响应滑动操作问题
2. 解决 UIScrollView 中添加 UITapGestureRecognizer 跟 UIButton 点击事件冲突

## 参考链接

1. [ITEYE - 为UILabel添加点击事件](#)
2. [怎么解决UIScrollView中添加UITapGestureRecognizer跟UIButton点](#)

## 详细内容

**UIScrollView \ UITableViewCell 添加 UIButton 时不响应滑动操作问题**

```
//1.添加单击事件
...
UILabel *label = [[UILabel alloc] initWithFrame:CGRectMake(60, 0
, listV.frame.size.width - 60, listV.frame.size.height)];
label.userInteractionEnabled=YES;
label.tag = 100;
UITapGestureRecognizer *labelTapGestureRecognizer = [[UITapGestureRecognizer alloc]initWithTarget:self action:@selector(labelTouchUpInside:)];
[label addGestureRecognizer:labelTapGestureRecognizer];
...

//2.获取单击事件
-(void) labelTouchUpInside:(UITapGestureRecognizer *)recognizer
{
    UILabel *label=(UILabel*)recognizer.view;
    NSLog(@"%@", -- Tag : %ld",label.text,label.tag);
}
```

**UIScrollView** 中添加 **UITapGestureRecognizer** 跟 **UIButton** 点击事件冲突

```
//1.添加单击事件
...
UITapGestureRecognizer *gensture = [[UITapGestureRecognizer alloc] initWithTarget:self action:@selector(scrollViewTapAction)];
gensture.delegate = self;
[scrollView addGestureRecognizer:gensture];
...

//2.实现 UIGestureRecognizerDelegate 中的代理方法

- (BOOL)gestureRecognizer:(UIGestureRecognizer *)gestureRecognizer shouldReceiveTouch:(UITouch *)touch
{
    UIView *view = [touch view];
    if ([view isKindOfClass:[UIButton class]])
    {
        return NO;
    }

    return YES;
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-27 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UIScrollView - 给 UIScrollView 添加 Autolayout 约束条件

## 关键字

UIScrollView \ Autolayout \ 自适应大小的 UIScrollView

## 需求场景

- 一些应用的详情页面需要使用 UIScrollView 时

## 参考链接

- [iOS: How To Make AutoLayout Work On A ScrollView](#)

## 详细内容

### 1. 已纵向滑动为例

- 首先，在需要滑动显示的页面 Xib 主 View 下拖放一个 UIScrollView，并设置约束条件为上下左右四个方向距离主 View 都为 0；
- 在 UIScrollView 中添加一个 UIView 作为 ContentView，并设置约束条件为上下左右四个方向距离主 UIScrollView 都为 0；
- 同时选中 UIScrollView 和 ContentView，通过 Editor -> Pin -> Widths Equally 设置宽度相等；

4. 最重要的一步，*ContentView* 的高度需要根据其内容约束计算出来，*ContentView* 中距离 *ContentView* 最底部的一个 *View* 的约束条件至少包括 "底部距离 *ContentView* XX 像素" 且自身高度可根据其他约束条件（与 *ContentView* 无关联的约束条件）计算出得出或直接设置得出。

**2.** 横向滑动设置步骤**2**的高度相等即可

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-23 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UIearchBar - 通过 UISearchDisplayViewController 实现全屏搜索显示效果

## 关键字

UIearchBar \ UISearchDisplayViewController \ 搜索 \ 查询 \ 全屏搜索

## 需求场景

1. 实现电机 SearchBar 进入全屏搜索模式的场景

## 参考链接

1. [CSDN - UISearchDisplayController学习笔记](#)

## 详细内容

1. 实例化 **UISearchDisplayViewController** 对象

```
@property (nonatomic, strong) UISearchDisplayController *searchController;

- (void)addSearchBarAndSearchDisplayController {

    UISearchBar *searchBar = [[UISearchBar alloc] initWithFrame:CGRectMakeZero];
    [searchBar sizeToFit];
    searchBar.delegate = self;

    self.tableView.tableHeaderView = searchBar;

    UISearchDisplayController *searchDisplayController = [[UISearchDisplayController alloc] initWithSearchBar:searchBar contentsController:self];
    searchDisplayController.delegate = self;
    searchDisplayController.searchResultsDataSource = self;
    searchDisplayController.searchResultsDelegate = self;

    self.searchController = searchDisplayController;
}
```

2. 进入全屏搜索模式，需要设置 **UISearchBar** 的 **delegate**，并实现 **searchBarShouldBeginEditing** 方法。在中间执行：

```
[self.searchDisplayController setActive:YES animated:YES];
```

3. 实现必要的代理方法

```
//在搜索内容改变时调用  
-(BOOL)searchDisplayController:(UISearchDisplayController *)controller shouldReloadTableForSearchString:(NSString *)searchString  
{  
    return YES;  
}  
  
//在搜索范围改变时调用(可选)  
- (BOOL)searchDisplayController:(UISearchDisplayController *)controller shouldReloadTableForSearchScope:(NSInteger)searchOption  
{  
    return YES;  
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-23 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UITableView - 使用 EGORefreshTableHeaderView 实现下拉刷新 UITableView

## 关键字

UITableView \ EGORefreshTableHeaderView \ 下拉刷新

## 需求场景

1. 需要实现列表数据的下拉刷新

## 参考链接

1. CSDN - IOS详解TableView——内置刷新，EGO，以及搜索显示控制器
2. GitHub - EGOTableViewPullRefresh

## 详细内容

### 1. EGORefreshTableHeaderView(.h.m)

```
1. EGORefreshTableHeaderView.h ````objectivec // //
EGORefreshTableHeaderView.h // Demo // // Created by Devin Doty on
10/14/09October14. // Copyright 2009 enorgemo. All rights reserved. // //
Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to deal
// in the Software without restriction, including without limitation the rights // to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell // copies
```

of the Software, and to permit persons to whom the Software is // furnished to do so, subject to the following conditions: // // The above copyright notice and this permission notice shall be included in // all copies or substantial portions of the Software. // // THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR // IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, // FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE // AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER // LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, // OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN // THE SOFTWARE. //

**import**

**import**

```
typedef enum{ EGOOPullPulling = 0, EGOOPullNormal, EGOOPullLoading, }  
EGOPullState;
```

**define DEFAULT\_ARROW\_IMAGE  
[UIImage imageNamed:@"blueArrow.png"]**

**define DEFAULT\_BACKGROUND\_COLOR  
[UIColor colorWithRed:226.0/255.0  
green:231.0/255.0 blue:237.0/255.0  
alpha:1.0]**

**define DEFAULT\_TEXT\_COLOR [UIColor  
colorWithRed:87.0/255.0 green:108.0/255.0**

**blue:137.0/255.0 alpha:1.0]**

**define**

**DEFAULT\_ACTIVITY\_INDICATOR\_STYLE  
UIActivityIndicatorViewStyleWhite**

**define FLIP\_ANIMATION\_DURATION 0.18f**

**define PULL\_AREA\_HEIGHT 60.0f**

**define PULL\_TRIGGER\_HEIGHT  
(PULL\_AREA\_HEIGHT + 5.0f)**

```
@protocol EGOResfreshTableHeaderDelegate; @interface  
EGOResfreshTableHeaderView : UIView {
```

```
    id _delegate;  
    EGOPullState _state;  
  
    UILabel *_lastUpdatedLabel;  
    UILabel *_statusLabel;  
    CALayer *_arrowImage;  
    UIActivityIndicatorView *_activityView;  
  
    // Set this to Yes when egoRefreshTableHeaderDidTriggerRefresh d  
    // elegate is called and No with egoRefreshScrollViewDataSourceDidF  
    // inishedLoading  
    BOOL isLoading;
```

}

```
@property(nonatomic,assign) id delegate;
```

- (void)refreshLastUpdatedDate;
- (void)egoRefreshScrollViewDidScroll:(UIScrollView \*)scrollView;
- (void)egoRefreshScrollViewDidEndDragging:(UIScrollView \*)scrollView;
- (void)egoRefreshScrollViewWillBeginDragging:(UIScrollView \*)scrollView;
- (void)egoRefreshScrollViewDataSourceDidFinishedLoading:(UIScrollView \*)scrollView;
- (void)startAnimatingWithScrollView:(UIScrollView \*) scrollView;
- (void)setBackgroundColor:(UIColor )*backgroundColor* textColor:(UIColor )*textColor* arrowImage:(UIImage \*)*arrowImage*;

@end

@protocol EGOResfreshTableHeaderDelegate

- (void)egoRefreshTableHeaderDidTriggerRefresh:(EGOResfreshTableHeaderView\*)view; @optional
- (NSDate)egoRefreshTableHeaderDataSourceLastUpdated:(EGOResfreshTableHeaderView)view;

@end

```
1. EGOResfreshTableHeaderView.m
```
objectivec
//
// EGOResfreshTableHeaderView.m
// Demo
//
// Created by Devin Doty on 10/14/09 October 14.
// Copyright 2009 enormego. All rights reserved.
//
// Permission is hereby granted, free of charge, to any person
obtaining a copy
// of this software and associated documentation files (the "Software"),
// to deal
// in the Software without restriction, including without limitation
// the rights
// to use, copy, modify, merge, publish, distribute, sublicense,
// and/or sell
// copies of the Software, and to permit persons to whom the So
```

```
ftware is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall
be included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KI
ND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERC
HANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGE
S OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWI
SE, ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHE
R DEALINGS IN
// THE SOFTWARE.
//


#import "EGOResfreshTableHeaderView.h"

@interface EGOResfreshTableHeaderView (Private)
- (void)setState:(EGOPullState)aState;
@end

@implementation EGOResfreshTableHeaderView

@synthesize delegate=_delegate;

- (id)initWithFrame:(CGRect)frame {
    if (self = [super initWithFrame:frame]) {

        isLoading = NO;

        CGFloat midY = frame.size.height - PULL_AREA_HEIGHT/2;

        /* Config Last Updated Label */
    }
}
```

```
UILabel *label = [[UILabel alloc] initWithFrame:CGRectMake(0.0f, midY, self.frame.size.width, 20.0f)];
    label.autoresizingMask = UIViewAutoresizingFlexibleWidth;
;
    label.font = [UIFont systemFontOfSize:12.0f];
    label.shadowOffset = CGSizeMake(0.0f, 1.0f);
    label.backgroundColor = [UIColor clearColor];
    label.textAlignment = NSTextAlignmentCenter;
    [self addSubview:label];
    _lastUpdatedLabel=label;

/* Config Status Updated Label */
label = [[UILabel alloc] initWithFrame:CGRectMake(0.0f,
midY - 18, self.frame.size.width, 20.0f)];
    label.autoresizingMask = UIViewAutoresizingFlexibleWidth;
;
    label.font = [UIFont boldSystemFontOfSize:13.0f];
    label.shadowOffset = CGSizeMake(0.0f, 1.0f);
    label.backgroundColor = [UIColor clearColor];
    label.textAlignment = NSTextAlignmentCenter;
    [self addSubview:label];
    _statusLabel=label;

/* Config Arrow Image */
CALayer *layer = [[CALayer alloc] init];
    layer.frame = CGRectMake(25.0f,midY - 35, 30.0f, 55.0f);
    layer.contentsGravity = kCAGravityResizeAspect;
#ifndef __IPHONE_OS_VERSION_MAX_ALLOWED >= 40000
    if ([[UIScreen mainScreen] respondsToSelector:@selector(scale)])
        layer.contentsScale = [[UIScreen mainScreen] scale];
}
#endif
[[self layer] addSublayer:layer];
_arroImage=layer;

/* Config activity indicator */
UIActivityIndicatorView *view = [[UIActivityIndicatorView view alloc] initWithFrame:CGRectMake(0.0f, midY - 35, 30.0f, 55.0f)];
    view.activityIndicatorStyle = UIActivityIndicatorViewStyleDefault;
    [self addSubview:view];
    _activityIndicatorView=view;
```

```
    view.frame = CGRectMake(25.0f,midY - 8, 20.0f, 20.0f);
    [self addSubview:view];
    _activityView = view;

    [self setState:EGOOPullNormal];

    /* Configure the default colors and arrow image */
    [self setBackgroundColor:nil textColor:nil arrowImage:nil];
}

return self;

}

#pragma mark -
#pragma mark Setters

#define aMinute 60
#define anHour 3600
#define aDay 86400

- (void)refreshLastUpdatedDate {
    NSDate * date = nil;
    if ([_delegate respondsToSelector:@selector(egoRefreshTableHeaderDataSourceLastUpdated:)]) {
        date = [_delegate egoRefreshTableHeaderDataSourceLastUpdated:self];
    }
    if(date) {
        NSTimeInterval timeSinceLastUpdate = [date timeIntervalSinceNow];
        NSInteger timeToDisplay = 0;
        timeSinceLastUpdate *= -1;

        if(timeSinceLastUpdate < anHour) {
            timeToDisplay = (NSInteger) (timeSinceLastUpdate / a
Minute);
        }
    }
}
```

```
        if(timeToDisplay == /* Singular */ 1) {
            _lastUpdatedLabel.text = [NSString stringWithFormat: NSLocalizedStringFromTable(@"更新于 %ld 分钟之前", @"PullTableViewLan", @"Last update in minutes singular"), (long)timeToDisplay];
        } else {
            /* Plural */
            _lastUpdatedLabel.text = [NSString stringWithFormat:NSLocalizedStringFromTable(@"更新于 %ld 分钟之前", @"PullTableViewLan", @"Last update in minutes plural"), (long)timeToDisplay];
        }

    }

} else if (timeSinceLastUpdate < aDay) {
    timeToDisplay = (NSInteger) (timeSinceLastUpdate / aHour);
    if(timeToDisplay == /* Singular */ 1) {
        _lastUpdatedLabel.text = [NSString stringWithFormat:NSLocalizedStringFromTable(@"更新于 %ld 小时之前", @"PullTableViewLan", @"Last update in hours singular"), (long)timeToDisplay];
    } else {
        /* Plural */
        _lastUpdatedLabel.text = [NSString stringWithFormat:NSLocalizedStringFromTable(@"更新于 %ld 小时之前", @"PullTableViewLan", @"Last update in hours plural"), (long)timeToDisplay];
    }

}

} else {
    timeToDisplay = (NSInteger) (timeSinceLastUpdate / aDay);
    if(timeToDisplay == /* Singular */ 1) {
        _lastUpdatedLabel.text = [NSString stringWithFormat:NSLocalizedStringFromTable(@"更新于 %ld 天之前", @"PullTableViewLan", @"Last update in days singular"), (long)timeToDisplay];
    } else {
        /* Plural */
        _lastUpdatedLabel.text = [NSString stringWithFormat:NSLocalizedStringFromTable(@"更新于 %ld 天之前", @"PullTableViewLan", @"Last update in days plural"), (long)timeToDisplay];
    }

}
```

```
wLan",@"Last update in days plural"), (long)timeToDisplay];
    }

}

} else {
    _lastUpdatedLabel.text = nil;
}

// Center the status label if the lastupdate is not available
CGFloat midY = self.frame.size.height - PULL_AREA_HEIGHT/2;
if(!_lastUpdatedLabel.text) {
    _statusLabel.frame = CGRectMake(0.0f, midY - 8, self.frame.size.width, 20.0f);
} else {
    _statusLabel.frame = CGRectMake(0.0f, midY - 18, self.frame.size.width, 20.0f);
}

}

- (void)setState:(EGOPullState)aState{

switch (aState) {
case EGOPullPulling:

    _statusLabel.text = NSLocalizedStringFromTable(@"Release to refresh ...",@"PullTableViewLan", @"Release to refresh status");
    [CATransaction begin];
    [CATransaction setAnimationDuration:FLIP_ANIMATION_DURATION];
    _arrowImage.transform = CATransform3DMakeRotation((M_PI / 180.0) * 180.0f, 0.0f, 0.0f, 1.0f);
    [CATransaction commit];

    break;
case EGOPullNormal:
```

```
        if (_state == EG00PullPulling) {
            [CATransaction begin];
            [CATransaction setAnimationDuration:FLIP_ANIMATION_DURATION];
            _arrowImage.transform = CATransform3DIdentity;
            [CATransaction commit];
        }

        _statusLabel.text = NSLocalizedStringFromTable(@"Pull down to refresh ...",@"PullTableViewLan", @"Pull down to refresh status");
        [_activityView stopAnimating];
        [CATransaction begin];
        [CATransaction setValue:(id)kCFBooleanTrue forKey:kCATransactionDisableActions];
        _arrowImage.hidden = NO;
        _arrowImage.transform = CATransform3DIdentity;
        [CATransaction commit];

        [self refreshLastUpdatedDate];

        break;
    case EG00PullLoading:

        _statusLabel.text = NSLocalizedStringFromTable(@"Loading ...",@"PullTableViewLan", @"Loading Status");
        [_activityView startAnimating];
        [CATransaction begin];
        [CATransaction setValue:(id)kCFBooleanTrue forKey:kCATransactionDisableActions];
        _arrowImage.hidden = YES;
        [CATransaction commit];

        break;
    default:
        break;
}

_state = aState;
```

```
- (void)setBackgroundColor:(UIColor *)backgroundColor textColor:(UIColor *) textColor arrowImage:(UIImage *) arrowImage
{
    self.backgroundColor = backgroundColor? backgroundColor : DEFAULT_BACKGROUND_COLOR;

    if(textColor) {
        _lastUpdatedLabel.textColor = textColor;
        _statusLabel.textColor = textColor;
    } else {
        _lastUpdatedLabel.textColor = DEFAULT_TEXT_COLOR;
        _statusLabel.textColor = DEFAULT_TEXT_COLOR;
    }
    _lastUpdatedLabel.shadowColor = [_lastUpdatedLabel.textColor colorWithAlphaComponent:0.1f];
    _statusLabel.shadowColor = [_statusLabel.textColor colorWithAlphaComponent:0.1f];

    _arrowImage.contents = (id)(arrowImage? arrowImage.CGImage : DEFAULT_ARROW_IMAGE.CGImage);
}

#pragma mark -
#pragma mark ScrollView Methods

- (void)egoRefreshScrollViewDidScroll:(UIScrollView *)scrollView
{
    if (_state == EGOOPullLoading) {
        CGFloat offset = MAX(scrollView.contentOffset.y * -1, 0)
    ;
        offset = MIN(offset, PULL_AREA_HEIGHT);
        UIEdgeInsets currentInsets = scrollView.contentInset;
        currentInsets.top = offset;
        scrollView.contentInset = currentInsets;

    } else if (scrollView.isDragging) {
        if (_state == EGOOPullPulling && scrollView.contentOffset.y > -PULL_TRIGGER_HEIGHT && scrollView.contentOffset.y < 0.0f
```

```
&& !isLoading) {  
    [self setState:EGOOPullNormal];  
} else if (_state == EGOOPullNormal && scrollView.contentOffset.y < -PULL_TRIGGER_HEIGHT && !isLoading) {  
    [self setState:EGOOPullPulling];  
  
}  
  
if (scrollView.contentInset.top != 0) {  
    UIEdgeInsets currentInsets = scrollView.contentInset;  
;  
    currentInsets.top = 0;  
    scrollView.contentInset = currentInsets;  
}  
  
}  
  
}  
  
- (void)startAnimatingWithScrollView:(UIScrollView *) scrollView  
{  
    isLoading = YES;  
  
    [self setState:EGOOPullLoading];  
    [UIView beginAnimations:nil context:NULL];  
    [UIView setAnimationDuration:0.2];  
    UIEdgeInsets currentInsets = scrollView.contentInset;  
    currentInsets.top = PULL_AREA_HEIGHT;  
    scrollView.contentInset = currentInsets;  
    [UIView commitAnimations];  
    if(scrollView.contentOffset.y == 0){  
        [scrollView setContentOffset:CGPointMake(scrollView.contentOffset.x, -PULL_TRIGGER_HEIGHT) animated:YES];  
    }  
}  
  
- (void)egoRefreshScrollViewDidEndDragging:(UIScrollView *)scrollView {  
  
    if (scrollView.contentOffset.y <= - PULL_TRIGGER_HEIGHT && !
```

```
isLoading) {
    if ([_delegate respondsToSelector:@selector(egoRefreshTableHeaderDidTriggerRefresh:)]) {
        [_delegate egoRefreshTableHeaderDidTriggerRefresh:sender];
    }
    [self startAnimatingWithScrollView:scrollView];
}

- (void)egoRefreshScrollViewDataSourceDidFinishLoading:(UIScrollView *)scrollView {
    isLoading = NO;

    [UIView beginAnimations:nil context:NULL];
    [UIView setAnimationDuration:.3];
    UIEdgeInsets currentInsets = scrollView.contentInset;
    currentInsets.top = 0;
    scrollView.contentInset = currentInsets;
    [UIView commitAnimations];

    [self setState:EGOOPullNormal];
}

- (void)egoRefreshScrollViewWillBeginDragging:(UIScrollView *)scrollView
{
    [self refreshLastUpdatedDate];
}

#pragma mark -
#pragma mark Deallocation

- (void)dealloc {
    _delegate=nil;
    [_activityView release];
}
```

```
[_statusLabel release];
[_arrowImage release];
[_lastUpdatedLabel release];
[super dealloc];
}

@end
```

## 2. 设置 ARC

```
Project 名称 -> TARGETS -> Build Phases -> Compile Sourccef
设置 EGOResfreshTableHeaderView.m 属性 -fno-objc-arc
```

## 3. 引入并使用

### 1. 引入头文件

```
#import "EGOResfreshTableHeaderView.h"
```

### 2. 实例化 EGOResfreshTableHeaderView 对象 `` swift var headerRefreshView : EGOResfreshTableHeaderView!

```
func setupRefreshTableView() { headerRefreshView =
EGOResfreshTableHeaderView(frame: CGRectMake(0, 0 -
CGRectGetHeight(tableViewMain.frame),
CGRectGetWidth(tableViewMain.frame),
CGRectGetHeight(tableViewMain.frame))) headerRefreshView.delegate = self
self.tableViewMain.addSubview(headerRefreshView) }
```

## 3. 实现必要的代理

```
```swift
EGORefreshTableHeaderDelegate 和 UIScrollViewDelegate

// MARK : - Drag Refresh

func egoRefreshTableHeaderDidTriggerRefresh(view: EGORefreshTableHeaderView!) {
    //
    println("egoRefreshTableHeaderDidTriggerRefresh")

    self.tableView.reloadData()

    dispatch_after(
        dispatch_time(
            DISPATCH_TIME_NOW,
            Int64(3.0 * Double(NSEC_PER_SEC)))
    ),
    dispatch_get_main_queue(), {
        self.endReloadTableViewDataSource()
    }
}

func scrollViewDidScroll(scrollView: UIScrollView) {
    headerRefreshView.egoRefreshScrollViewDidScroll(scrollView)
}

func scrollViewDidEndDragging(scrollView: UIScrollView, willDecelerate decelerate: Bool) {
    headerRefreshView.egoRefreshScrollViewDidEndDragging(scrollView)
}
```

## 1. 实现刷新请求和结束操作函数

{

```
func endReloadTableViewDataSource() {  
    self.headerRefreshView.egoRefreshScrollViewDataSourceDidFinishedLoading(sel  
f.tableViewMain) } ````
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-09-08 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UITableView - 滑动加载性能优化总结

## 关键字

UITableView \ UITableViewCell \ reloadData \ 列表 \ 滑动 \ 卡顿 \ 性能优化

## 需求场景

- 实现较为复杂的 UITableViewCell 列表和加载大量数据时

## 参考链接

1. [CocoaChina - 一次 TableView 性能优化经历](#)
2. [伯乐在线 - iOS应用性能调优的25个建议和技巧](#)
3. [iOS App性能优化](#)
4. [GitHub - UITableView优化技巧](#)
5. [UITableView 滚动流畅性优化](#)

## 详细内容

1. 列表卡顿问题最好真机测试，有条件的尽量选择低版本硬件和系统进行测试；
2. 使用 **Instruments** 的 **Time Profiler** 工具定位造成卡顿时间消耗的位置；
3. 避免 **UITableView** 的多次刷新(**reloadData**)，尤其 **Xib** 加载 **UITableView** 时避免首次自动加载；

**4. 为 Cell 专门定义显示 Model :**

**5. Model 需要包含已提前计算出的 Cell 高度；**

**6. 对于显示的 NSString，提前在 Model 中组装完成，避免在 Cell 中组装转换；**

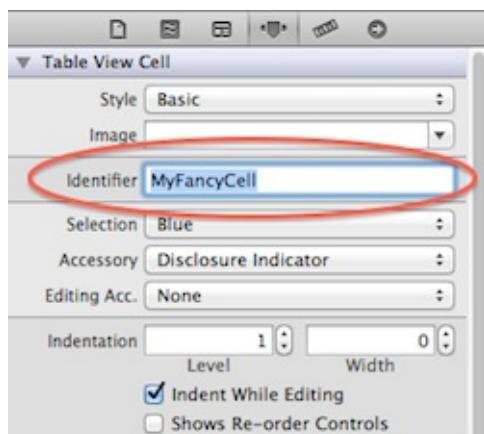
**7. 对于需要加载的网络图片链接，提前在 Model 中组装完成 NSURL，避免在 Cell 中组装转换；**

**8. 尽量减少 Cell 中的逻辑判断和运算并使用正确的数据格式；**

**9. 避免在 Cell 中反复创建 View，最好在初始化时一并创建，通过设置 Hidden 属性控制显示和隐藏；**

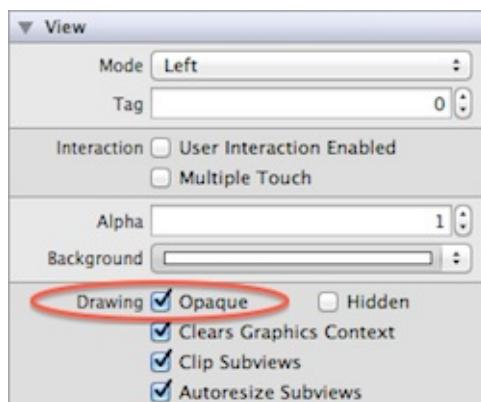
**10. 对于 UIImageView，注意加载的图片大小是否与控件大小一致，尽量保持一致；**

**11. 正确使用 reuseIdentifier**

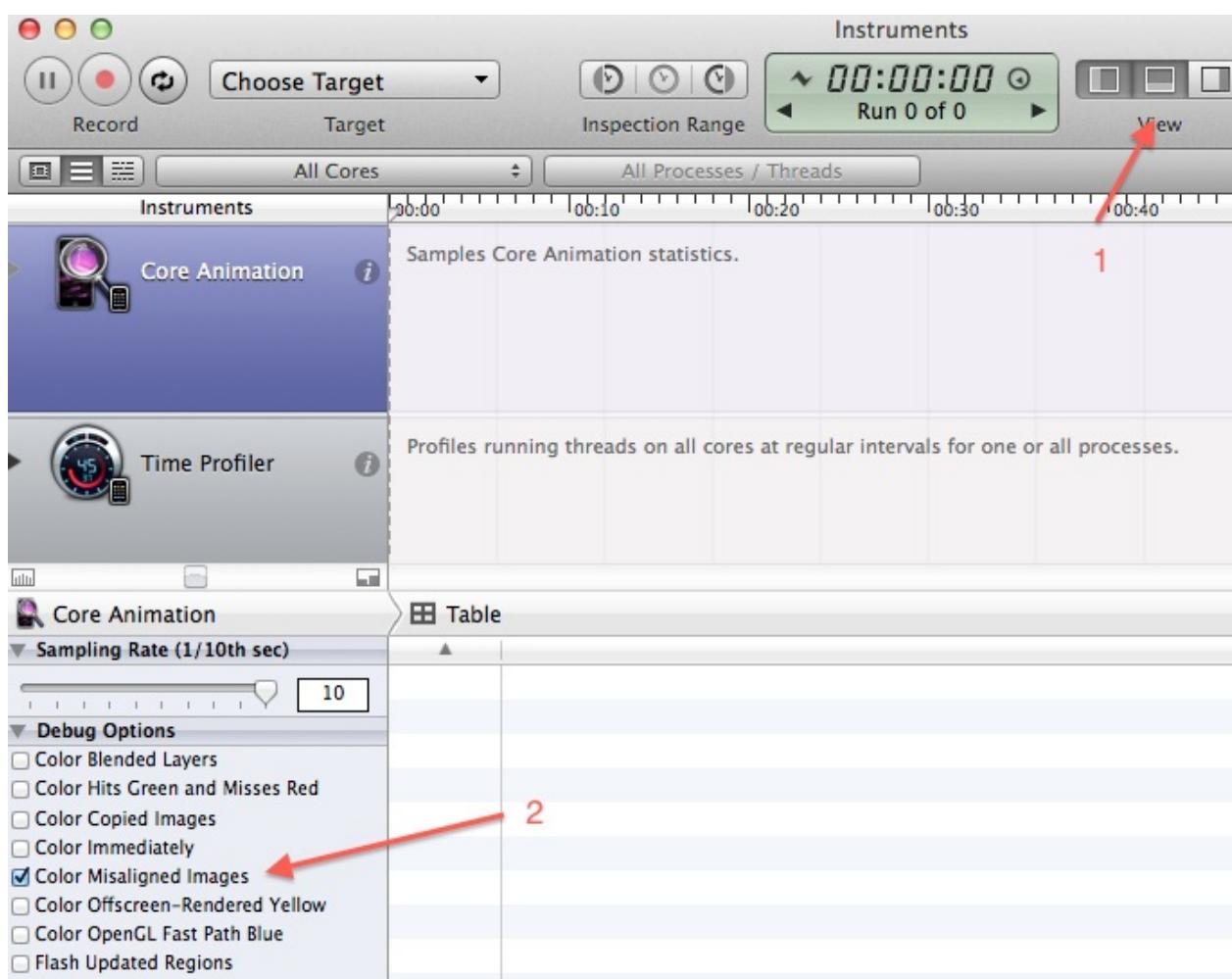


```
static NSString *CellIdentifier = @"MyFancyCell";
UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier forIndexPath:indexPath];
```

**12. 尽量把所有的 UIView 以及其子类控件设置为不透明（Opaque = YES），你可以在模拟器中用 Debug\Color Blended Layers 选项来发现哪些 view 没有被设置为 opaque。目标就是，能设为 opaque 的就全设为 opaque!**



**13.** 尽量避免图片压缩，模拟器条件下使用 **Debug\Color Misaligned Images** 选项检查像素是否对其：如果图片边界没有与目标像素完美对齐，该功能可为图片叠加一层品红色。如果图片使用确定的比例大小绘制，那么该功能会为图片添加一层黄色叠加。真机条件下使用 **Instrument** 中的 **Core Animation** 工具检查；



**14.** 使用 **shadowPath** 来画阴影；

```
#import <QuartzCore/QuartzCore.h>

// Somewhere later ...
UIView *view = [[UIView alloc] init];

// Setup the shadow ...
view.layer.shadowOffset = CGSizeMake(-1.0f, 1.0f);
view.layer.shadowRadius = 5.0f;
view.layer.shadowOpacity = 0.6;
```

**15.** 减少 **subviews** 的数量；

**16.** 尽量不使用 **cellForRowAtIndexPath:**，如果你需要用到它，只用一次然后缓存结果；

**17.** 使用 **Autorelease Pool**：

```
NSArray *urls = <# An array of file URLs #>;
for (NSURL *url in urls) {
    @autoreleasepool {
        NSError *error;
        NSString *fileContents = [NSString stringWithContentsOfU
RL:url
                           encoding:NSUTF8StringEncoding
                           error:&error];
    }
}
```

**18.** 选择是否需要缓存；

```
UIImage *img = [UIImage imageNamed:@"myImage"];
// caching

// or
UIImage *img = [UIImage imageWithContentsOfFile:@"myImage"];
// no caching
```

如果你要加载一个大图片而且是一次性使用，那么就没必要缓存这个图片，用 *imageWithContentsOfFile* 足矣，这样不会浪费内存来缓存它。然而，在图片反复重用的情况下  *imageNamed* 是一个好得多的选择。

**19.** 避免日期格式转换；相关日期显示应该在 **Model** 中定义完成。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-24 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

## 关键字

## 需求场景

1. UITableView 中第一个 cell 和最后一个 cell 距离顶部和底部距离差异

## 参考链接

1. [CocoaChina - tableView的第一个cell到背景顶部的距离](#)

## 详细内容

设置方法如下

```
self.tableView.contentInset = UIEdgeInsets(top: 8, left: 0, bottom: 8, right: 0)
```

未设置效果

Carrier 3:01 PM

All Available My Bidding Favorite

|                              |             |        |
|------------------------------|-------------|--------|
| Auction Name A<br>My rank 99 | 00h 55m 11s |        |
|                              | Lowest Bid  | My Bid |
|                              | \$160       | \$180  |

|                              |             |        |
|------------------------------|-------------|--------|
| Auction Name A<br>My rank 99 | 00h 55m 11s |        |
|                              | Lowest Bid  | My Bid |
|                              | \$160       | \$180  |

|                              |             |        |
|------------------------------|-------------|--------|
| Auction Name A<br>My rank 99 | 00h 55m 11s |        |
|                              | Lowest Bid  | My Bid |
|                              | \$160       | \$180  |

设置后效果

Carrier WiFi

3:04 PM

**All Available**

My Bidding

Favorite

**Auction Name A**

My rank 99

🕒 00h 55m 11s



Lowest Bid

\$160

My Bid

\$180

**Auction Name A**

My rank 99

🕒 00h 55m 11s



Lowest Bid

\$160

My Bid

\$180

**Auction Name A**

My rank 99

🕒 00h 55m 11s



Lowest Bid

\$160

My Bid

\$180

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-01 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UITableViewCell - 动态修改 UITableViewCell 高度

## 关键字

UITableViewCell \ UITableView \ 高度 \ 动态调整高度

## 需求场景

1. 需要根据不同内容动态显示 Cell 高度时

## 参考链接

1. [动态计算UITableViewCell高度详解](#)
2. [使用Autolayout实现UITableView的Cell动态布局和高度动态改变](#)

## 详细内容

### 1. 基本原理

1. 通过对 UITableViewCell 中显示文字相关控件大小的计算，得出理想的显示控件大小；
2. 根据显示控件大小，计算出合适的 Cell 高度；
3. 在 UITableView reloadData 时，通过 UITableView 的 heightForRowAtIndexPath 方法返回合适的 Cell 高度。

### 2. 示例

## 1. Swift 示例

示例一：使用 sizeThatFits

```

var itemCell : UITableViewCell?

itemCell = tableViewMain.dequeueReusableCell(withIdentifier("REXP
rojectInfoItemCell") as? REXProjectInfoItemCell

func tableView(tableView: UITableView, heightForRowAtIndexPath i
ndексPath: NSIndexPath) -> CGFloat {

    var cell :REXProjectInfoItemCell = self.itemCell as REXProje
ctInfoItemCell

    cell.labelTitle.text = "Auction Invite Welcome Message (Text
Note)"
    cell.labelInfo.text = "You've been invited to participate in
the upcoming auction for the SAP Sales & Distribution Lead....."

    var s : CGSize = cell.labelInfo.sizeThatFits(CGSizeMake(WI
DTHSCREEN - 50, CGFloat(FLOAT_MAX)))
    var defaultHeight : CGFloat = cell.contentView.frame.size.h
eight
    var height : CGFloat = s.height + 36 > defaultHeight ? s.hei
ght + 36 : defaultHeight

    return 1 + height
}

```

注意：

1. 用于显示较长文字的控件 `labelInfo` 需要设置 `Lines` 属性为 0；设置 `Line Breaks` 属性为 `Word Wrap`；这样才能正确换行
2. `sizeThatFits` 函数的参数是限宽不限高（用于计算高度）或者限高不限宽（用于计算宽度）的，所限制的宽或者高根据页面布局需求设定
3. 避免因为显示文字内容过短或为空导致计算高度过小，需要设置 `defaultHeight`，`defaultHeight` 可根据自定义 Cell 的默认 Frame 计算出。

4. 示例中的 36 是 padding 值，即除了显示文字控件自己高度外，控件本身距离 Cell 上下边界的高度。
5. 最后的 +1 操作是因为默认的 Cell 高度要比 contentView 高度高 1 个像素，具体可查看自定义 Cell 的 Xib 文件。为了正常显示，需要执行 +1 操作。

示例二：使用 systemLayoutSizeFittingSize

```
var receiveCell: REXMessageReceiveReplyCell!

tableViewMain.registerNib(UINib(nibName: "REXBidHistoryItemCell"
, bundle: nil), forCellReuseIdentifier: "REXBidHistoryItemCell")
receiveCell = tableViewMain.dequeueReusableCell(withIdentifier("REXMessageReceiveReplyCell") as? REXMessageReceiveReplyCell

func tableView(tableView: UITableView, heightForRowAtIndexPath indexPath: NSIndexPath) -> CGFloat {
    let message: Message = messagesList objectAtIndex(indexPath.row) as Message
    receiveCell.labelInfo.text = message.mInfo
    let size = receiveCell.contentView.systemLayoutSizeFittingSize(UILayoutFittingCompressedSize)
    return size.height + 1 < 65.0 ? 65.0 : size.height + 1
}
```

注意：

1. labelInfo 距离左右端约束中，可变的那一端设置约束条件为  $\geq$  或  $\leq$ ，不可变的那一端约束条件设置为  $=$
2. 避免因为显示文字内容过短或为空导致计算高度过小，需要设置 defaultHeight，defaultHeight 可根据自定义 Cell 的默认 Frame 计算出。以上示例中 defaultHeight 为 65.0
3. labelInfo 作为主要的约束参考条件，避免外部默认约束影响 labelInfo 的默认约束，比如 UIImageView

## 1. Objective-C 示例

1. 声明一个存计算 Cell 高度的实例变量

```
@property (nonatomic, strong) UITableViewCell *prototypeCell;
;
```

## 2. 初始化它

```
self.prototypeCell = [self.tableView dequeueReusableCellWithIdentifier:@"C1"];
```

## 3. 以下是实现部分

```
objective //注册自定义 Cell UINib *cellNib = [UINib
nibWithNibName:@"C1" bundle:nil]; [self.tableView registerNib:cellNib
forCellReuseIdentifier:@"C1"];
```

```
//初始化显示数据 self.tableData = @[@[@"1\n2\n3\n4\n5\n6",
@"123456789012345678901234567890", @"1\n2", @"1\n2\n3", @"1"]];
```

//实现必要的 UITableViewDelegate 和 UITableViewDataSource

- (UITableViewCell \*)tableView:(UITableView \*)tableView
cellForRowAtIndexPath:(NSIndexPath \*)indexPath { C5 cell = [self.tableView
dequeueReusableCellWithIdentifier:@"C5"]; cell.t.text = @"123";
cell.t.delegate = self; return cell; }
- (NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:
NSInteger)section { // Return the number of rows in the section. return
self.tableData.count; }
- (UITableViewCell \*)tableView:(UITableView \*)tableView
cellForRowAtIndexPath:(NSIndexPath \*)indexPath { C1 cell = [self.tableView
dequeueReusableCellWithIdentifier:@"C1"]; cell.t.text = [self.tableData
objectAtIndex:indexPath.row]; return cell; }

//下面是计算高度的代码

## pragma mark - UITableViewDelegate

- (CGFloat)tableView:(UITableView \*)tableView heightForRowAtIndexPath:
NSIndexPath \*)indexPath { C5 cell = (C5 )self.prototypeCell; cell.t.text =

```
self.updatedStr; CGSize s = [cell.t
sizeThatFits:CGSizeMake(cell.t.frame.size.width, FLT_MAX)]; CGFloat
defaultHeight = cell.contentView.frame.size.height; CGFloat height = s.height
> defaultHeight ? s.height : defaultHeight; return 1 + height; }
```

## pragma mark - UITextViewDelegate

- (BOOL)textView:(UITextView )*textView* shouldChangeTextInRange:(NSRange)*range* replacementText:(NSString )*text* { if ([*text* isEqualToString:@"\n"]) {

```
NSLog(@"h=%f", textView.contentSize.height);
```

} return YES; }
- (void)textViewDidChange:(UITextView \*)*textView* { self.updatedStr =
*textView*.text; [*self.tableView* beginUpdates]; [*self.tableView* endUpdates]; } `

## 效果图

(无)

## 备注

1. UILabel 的 preferredMaxLayoutWidth 属性会严重影响高度的计算，确保该属性在不同尺寸屏幕上均设置为正确的参数。
2. Select the Tighten Letter Spacing (adjustsLetterSpacingToFitWidth) checkbox if you want the spacing between letters to be adjusted to fit the string within the label's bounding rectangle. [参考链接](#) `swift import UIKit

```
class REXInfoLabel: UILabel {
```

```
override func layoutSubviews() {
    self.preferredMaxLayoutWidth = UIScreen.mainScreen().applicationFrame.size.width - 60
    super.layoutSubviews()
}

import UIKit

class REXHelpCell: UITableViewCell {

    @IBOutlet weak var labelQuestion: REXInfoLabel!
    @IBOutlet weak var labelAnswer: REXInfoLabel!

    override func awakeFromNib() {
        super.awakeFromNib()

        self.labelQuestion.layoutSubviews()
        self.labelAnswer.layoutSubviews()
    }
}

func tableView(tableView: UITableView, heightForRowAtIndexPath indexPath: NSIndexPath) -> CGFloat {
```

```
var receiveCell : REXHelpCell = REXHelpCell.loadFromNibNamed("REXHelpCell", bundle: nil) as REXHelpCell

var help : HelpItem = helpItems.objectAtIndex(indexPath.row) as HelpItem

receiveCell.labelQuestion.text = help.strQuestion

var height : CGFloat = 0.0
var defaultHeight : CGFloat = 0

if help.isOpen
{
    receiveCell.labelAnswer.text = help.strAnswer
}
else
{
    receiveCell.labelAnswer.text = ""
}

let size = receiveCell.contentView.systemLayoutSizeFittingSize(UILayoutFittingCompressedSize)

return size.height + 1 < defaultHeight ? defaultHeight : size.height + 1

}'''
```

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-28 | Alfred Jiang | -  |

## 方案名称

UITextField - 使用备忘

## 关键字

UITextField \ 使用备忘

## 需求场景

1. 记录使用过程中的备忘

## 参考链接

1. CSDN - iOS开发笔记--设置UITextField键盘上return key不可点击
2. Stack Overflow - Xcode UITextField limit type of characters
3. Stack Overflow - How to limit character input in UIAlertView UITextField

## 详细内容

### 1. 设置 **return key** 不可点击 (类似 **UISearchBar** 效果)

```
textField.returnKeyType = UIReturnKeySearch;           //设置按键类型为
Search 类型，return key 变为 "Search" 或 "搜索"
textField.enablesReturnKeyAutomatically = YES;        //输入框无文字
时 return key 灰色不可点
```

### 2. 限制输入指定长度 (**MAXLENGTH = 20**)

```
#define MAXLENGTH 20

- (BOOL)textField:(UITextField *)textField shouldChangeCharacter
sInRange:(NSRange)range replacementString:(NSString *)string
{
    if (![textField markedTextRange] && [textField.text length]
> MAXLENGTH) {

        textField.text = [textField.text substringToIndex:MAXLEN
GTH];

        return NO;
    }

    return YES;
}
```

以上方法在中文识别时有问题，建议修改为以下写法

```
[_textFieldInput addTarget:self action:@selector(textFieldDidChange:)
forControlEvents:UIControlEventEditingChanged]; //在 vie
wDidLoad 中注册 UIControlEventEditingChanged 监听事件

....
```

```
#define MAXLENGTH 20

- (void)textFieldDidChange:(UITextField *)textField
{
    if (![textField markedTextRange] && [textField.text length]
> MAXLENGTH) {

        textField.text = [textField.text substringToIndex:MAXLEN
GTH];
    }
}
```

### 3. 限制输入指定字符

```
#define ALPHABATES @“ABCDabcd”

- (BOOL)textField:(UITextField *)textField shouldChangeCharacter
sInRange:(NSRange)range replacementString:(NSString *)string
{
    NSCharacterSet *cs = [[NSCharacterSet characterSetWithCharac
tersInString:ALPHABATES] invertedSet];
    NSString *filtered = [[string componentsSeparatedByCharacter
sInSet:cs] componentsJoinedByString:@“”];
    return [string isEqualToString:filtered];
}
```

### 4. 取取消单词自动纠错

```
textField.autocorrectionType = UITextAutocorrectionTypeNo; //用
于输入英文 id 时，避免误纠正
```

### 5. 适配中英文输入时的字符长度监听（避免使用 **shouldChangeCharactersInRange** 出现删减时中文识别错误）

```
[_textFieldInput addTarget:self action:@selector(textFieldDidChange:)
forControlEvents:UIControlEventEditingChanged]; //在 vie
wDidLoad 中注册 UIControlEventEditingChanged 监听事件

....
```

```
- (void)textFieldDidChange:(UITextField *)textField
{
    NSLog(@"%@", [textField.text length]); //此时为 shouldChange
    CharactersInRange 执行后的正确长度
}
```

## 效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-09 | Alfred Jiang | -  |

## 方案名称

UITextView - 添加内容后自动滚动到最后一行

## 关键字

UITextView \ 动态添加 \ 自动滚动

## 需求场景

- 向 UITextView 中动态添加内容并希望自动滚动到底部时

## 参考链接

- CSDN - UITextView 内容添加后，自动滚动到最后一行

## 详细内容

- 设置 **layoutManager.allowsNonContiguousLayout** 属性为 **NO**

```
self.textViewInfo.layoutManager.allowsNonContiguousLayout = NO;
//这句代码设置了 UITextView 中的 layoutManager(NSLayoutManager) 的是否非连续布局属性，默认是 YES，设置为 NO 后 UITextView 就不会再自己重置滑动了。
```

- 执行 **scrollRangeToVisible** 方法，滑动到底部

```
[self.textViewInfo scrollRangeToVisible:NSMakeRange(0, textViewInfo.text.length)];
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-08-27 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

UIView - iOS 完全复制一个 UIView

## 关键字

UIView \ 复制 \ Duplicate

## 需求场景

1. 需要对一个 UIView 或其子类进行完整复制的操作

## 参考链接

1. [CSDN - iOS 完全复制 UIView](#)

## 详细内容

```
// Duplicate UIView
- (UIView*)duplicate:(UIView*)view
{
    NSData * tempArchive = [NSKeyedArchiver archivedDataWithRootObject:view];
    return [NSKeyedUnarchiver unarchiveObjectWithData:tempArchive];
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-17 | Alfred Jiang | -  |
| 2  | 2015-05-07 | Alfred Jiang | -  |
| 3  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

UIView - 代码实现截图功能

## 关键字

UIView \ 代码截图 \ 截图 \ 页面图片截取

## 需求场景

1. 需要在应用内对某个 View 进行截图操作时

## 参考链接

1. [CSDN - IOS开发之—程序截图](#)
2. [Stack Overflow - Screenshot in swift iOS?](#)

## 详细内容

1. 经典 **Objective-C** 解决方案

```
//获得View图像
- (UIImage *)imageFromView:(UIView *)theView
{
    UIGraphicsBeginImageContext(theView.frame.size);
    CGContextRef context = UIGraphicsGetCurrentContext();
    [theView.layer renderInContext:context];
    UIImage *theImage = UIGraphicsGetImageFromCurrentImageContext();
    UIGraphicsEndImageContext();

    return theImage;
}

//获得View某个范围内的图像
- (UIImage *)imageFromView:(UIView *)theView atFrame:(CGRect)r
{
    UIGraphicsBeginImageContext(theView.frame.size);
    CGContextRef context = UIGraphicsGetCurrentContext();
    CGContextSaveGState(context);
    UIRectClip(r);
    [theView.layer renderInContext:context];
    UIImage *theImage = UIGraphicsGetImageFromCurrentImageContext();
    UIGraphicsEndImageContext();

    return theImage;
}
```

## 2. iOS7 + Swift 解决方案

```

//  

//  UIView+Screenshot.swift  

//  VideoHouse  

//  

//  Created by gxw on 14/10/26.  

//  Copyright (c) 2014年 b-star. All rights reserved.  

//  
  

//实例一  

extension UIView {
    func screenshot() -> UIImage {
        var imageFrame = CGRectMake(0, 0, self.frame.size.width,
self.frame.height)
        UIGraphicsBeginImageContextWithOptions(imageFrame.size,
false, 0)
        self.drawViewHierarchyInRect(imageFrame, afterScreenUpdates: true)
        var screenshot = UIGraphicsGetImageFromCurrentImageContext()
        UIGraphicsEndImageContext()
        return screenshot
    }
}  
  

//实例二  

func screenShotMethod() {
    //Create the UIImage
    UIGraphicsBeginImageContext(view.frame.size)
    view.layer.renderInContext(UIGraphicsGetCurrentContext())
    let image = UIGraphicsGetImageFromCurrentImageContext()
    UIGraphicsEndImageContext()  
  

    //Save it to the camera roll
    UIImageWriteToSavedPhotosAlbum(image, nil, nil, nil)
}

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-08 | Alfred Jiang | -  |

## 方案名称

UIView - 代码设置 UIView 的阴影、圆角、边框效果

## 关键字

UIView \ 圆角 \ 阴影 \ 边框

## 需求场景

1. 需要为 UIView 等控件设置圆角、阴影、边框效果时

## 参考链接

1. 简书 - iOS给Ulview 加阴影加圆角加边框
2. CSDN - iPhone之为UIView设置阴影(CALayer的shadowColor，shadowOffset，shadowOpacity，shadowRadius,shadowPath属性)

## 详细内容

1. 导入 **#import**

2. 设置阴影

```
myview.layer.shadowOpacity = 0.5;// 阴影透明度  
  
myview.layer.shadowColor = [UIColor grayColor].CGColor;// 阴影的  
颜色  
  
myview.layer.shadowRadius = 3;// 阴影扩散的范围控制  
  
myview.layer.shadowOffset = CGSizeMake(1, 1); // 阴影的范围
```

### 3. 设置圆角

```
myview.layer.cornerRadius = 8;  
  
myview.layer.masksToBounds = YES;
```

### 4. 设置边框

```
myview.layer.borderWidth = 8;  
  
myview.layer.borderColor = [[UIColor grayColor] CGColor];
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-10-26 | Alfred Jiang | -  |
| 2  | 2015-12-18 | Alfred Jiang | -  |

## 方案名称

UIView \ UIButton - 独占响应事件

## 关键字

UIView \ UIButton \ 独占响应事件 \ 同时点击

## 需求场景

- 避免两个按钮（UIButton）很近的时候同时触发事件
- 避免同时点击多个按钮（UIButton）

## 参考链接

- IOS 开发两个按钮（button）很近的时候同时触发事件解决的方法  
(setExclusiveTouch)
- UIView独占响应事件

## 详细内容

```
_button1 = [[UIButton alloc] initWithFrame:CGRectMake(0, 0, 100, 100)];  
[self.view addSubview:_button1];  
_button1.backgroundColor = [UIColor redColor];  
[_button1 addTarget:self  
    action:@selector(buttonsEvent:)  
    forControlEvents:UIControlEventTouchUpInside];  
  
_button1.exclusiveTouch = YES;  
  
_button2 = [[UIButton alloc] initWithFrame:CGRectMake(0, 100, 100, 100)];  
[self.view addSubview:_button2];  
_button2.backgroundColor = [UIColor greenColor];  
[_button2 addTarget:self  
    action:@selector(buttonsEvent:)  
    forControlEvents:UIControlEventTouchUpInside];  
  
_button2.exclusiveTouch = YES;
```

## 效果图

(无)

## 备注

- **UIButton - 避免多次重复点击**

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-10-13 | Alfred Jiang | -  |
| 2  | 2015-12-18 | Alfred Jiang | -  |

## 方案名称

UIViewController - 旋转问题 willRotateToInterfaceOrientation 方法无法正常调用

## 关键字

UIViewController \ NavigationController \ 旋转

## 需求场景

1. iPad 端某些 View 无法正常旋转

## 参考链接

1. Stack Overflow - [willRotateToInterfaceOrientation not being called from presented viewController](#)

## 详细内容

可能原因：

1. iOS SDK 版本不对应， willRotateToInterfaceOrientation 方法在 iOS 8 以上已弃用。此时通过实现对应版本的推荐方法即可。
2. 没有正确的 addChildViewController 。对数原因是这个，下面参考 [Stackoverflow](#) 梳理解决方法

## 解决方法

1. 当添加 **NavigationController** 类容器作为子容器时，或者通过 **addSubview** 方法加载某 **UIViewController** 的子 **View** 时，若该容器或 **UIViewController** 需要接受旋转事件，那么需要调用 **addChildViewController:** 方法，以确定旋转事件可以继续向下传递；

```
[rootViewController addChildViewController:viewController];
```

## 2. 注册通知

1. 确保 **ViewDidLoad** 方法实现了 **[super ViewDidLoad]** 方法；
2. 在 **ViewDidLoad** 中注册通知

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(orientationChanged:) name:UIDeviceOrientationDidChangeNotification object:nil];
```

## 3. 实现消息接受

```
- (void)orientationChanged:(NSNotification *)notification{
    [self handleOrientation:[[UIApplication sharedApplication] statusBarOrientation]];
}
```

## 4. 实现对应处理 ``objectivec

```
5. (void) handleOrientation:(UIInterfaceOrientation) orientation {

if (orientation == UIInterfaceOrientationPortrait || orientation ==
UIInterfaceOrientationPortraitUpsideDown) {

    //handle the portrait view

} else if (orientation == UIInterfaceOrientationLandscapeLeft || orientation ==
UIInterfaceOrientationLandscapeRight) {

    //handle the landscape view

}}
```

6. 别忘了在 dealloc 方法中移除监听

```
[[NSNotificationCenter defaultCenter] removeObserver:self name:UIInterfaceOrientationDidChangeNotification object:nil];
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-09-09 | Alfred Jiang | -  |
| 2  | 2015-12-18 | Alfred Jiang | -  |

## 方案名称

UIWebView - 使用 UIWebViewToFile 实现 UIWebView 内容转为 Image 或 PDF

## 关键字

UIWebView \ UIImage \ PDF \ 长页面截图

## 需求场景

1. 需要将较长页面的 Web 内容转为 Image 或 PDF 文件时

## 参考链接

1. [GitHub - UIWebViewToFile](#)
2. [将UIWebView显示的内容转为图片和PDF](#)

## 详细内容

1. 添加头文件

```
#import "UIWebView+ToFile.h"
```

2. 调用公共接口函数实现转换 ``objectivec @interface UIWebView (ToFile)
 3. (UIImage \*)imageRepresentation;
 4. (NSData \*)PDFData;

@end ``

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-21 | Alfred Jiang | -  |

## 方案名称

Xcode - iOS 开发崩溃(Crash)调试

## 关键字

Xcode \ 崩溃 \ 日志 \ Crash \ Log \ dSYM \ Symbolicatecrash \ 调试

## 需求场景

1. 需要对 .crash 文件进行分析时

## 参考链接

1. 博客园 - iphone崩溃日志分析工具symbolicatecrash用法
2. CSDN - iOS通过dSYM文件分析crash
3. 简书 - iOS开发技巧—崩溃调试 - 崩溃调试的使用和技巧(推荐阅读)
4. GitHub - dSYMTools
5. CocoaChina - 如何使用symbolicatecrash工具

## 详细内容

### Xcode 6.0 之后版本快速使用：

在终端依次输入以下两条命令

```
export DEVELOPER_DIR="/Applications/XCode.app/Contents/Developer"
/Applications/Xcode.app/Contents/SharedFrameworks/DTDeviceKitBase.framework/Versions/A/Resources/symbolicatecrash xxx.crash xxx.app.dSYM
>crash.txt
```

- 所需文件：xxx.app xxx.app.dSYM xxx.crash

## 详细使用步骤介绍

### 1. 找到 **Symbolicatecrash** 文件

Xcode 6.0 之后

```
/Applications/Xcode.app/Contents/SharedFrameworks/DTDeviceKitBase.framework/Versions/A/Resources/
```

Xcode 6.0 之前

```
/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/Library/PrivateFrameworks/DTDeviceKit.framework/Versions/A/Resources/
```

Xcode 4.3 之前

```
/Developer/Platforms/iPhoneOS.platform/Developer/Library/PrivateFrameworks/DTDeviceKit.framework/Versions/A/Resources/
```

(附：Mac 系统显示隐藏文件

终端中输入以下命令

显示 Mac 隐藏文件的命令：defaults write com.apple.finder AppleShowAllFiles -bool true

隐藏 Mac 隐藏文件的命令：defaults write com.apple.finder AppleShowAllFiles -bool false

输入完回车，重启 Finder：左上角的苹果标志-->强制退出-->Finder-->重新启动 )

### 2. **Symbolicatecrash** 文件独立于 **Xcode**，可以拷出来使用

### 3. 命终端中输入命令，命令格式：**Symbolicatecrash .crash .dSYM > aa.log**

即：Symbolicatecrash + 崩溃日志 + APP 对应的 .dSYM 文件 + > + 输出到的文件

注意：symbolicatecrash的 参数顺序，否则会报错

#### 4. 如果提示"**DEVELOPER\_DIR**" is not defined

Xcode4.3之后 在终端中输入： export

DEVELOPER\_DIR=/Applications/Xcode.app/Contents/Developer

Xcode4.3之前 在终端中输入： export

DEVELOPER\_DIR=/Applications/Xcode.app

建议：每次打出版本后备份相应的.dSYM文件，对应版本崩溃后可根据此文件分析崩溃日志

更多内容见参考链接

### 效果图

(无)

### 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-10-09 | Alfred Jiang | -  |

## 方案名称

Xcode - Xcodeproj 介绍

## 关键字

Xcode \ Xcodeproj 介绍 \ 工程目录结构 \ 工程组织结构

## 需求场景

1. 需要了解 Xcode 工程组织结构时
2. 需要解决 Xcode Xcodeproj 文件冲突时

## 参考链接

1. [TOMORJM - 通过Xcodeproj深入探究Xcode工程文件 一](#)
2. [TOMORJM - 通过Xcodeproj深入探究Xcode工程文件 二](#)
3. [玉令天下的博客 - Let's Talk About project.pbxproj](#)
4. [GitHub - yulingtianxia/pbxprojHelper](#)

## 详细内容

(见参考链接)

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-14 | Alfred Jiang | -  |

## 方案名称

Xcode - 为 Xcode 添加 Snippets

## 关键字

Xcode \ Snippets

## 需求场景

1. 提高编程效率，添加代码 Snippets

## 参考链接

1. GitHub - Objective-C
2. GitHub - XcodeSwiftSnippets(推荐)
3. NSHipster - Xcode Snippets(推荐)

## 详细内容

(见参考链接)

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-15 | Alfred Jiang | -  |

## 方案名称

Xcode - 使用 BBUncrustifyPlugin-Xcode 插件实现代码格式化

## 关键字

Xcode \ BBUncrustifyPlugin \ 代码格式化

## 需求场景

1. 实现统一风格的代码

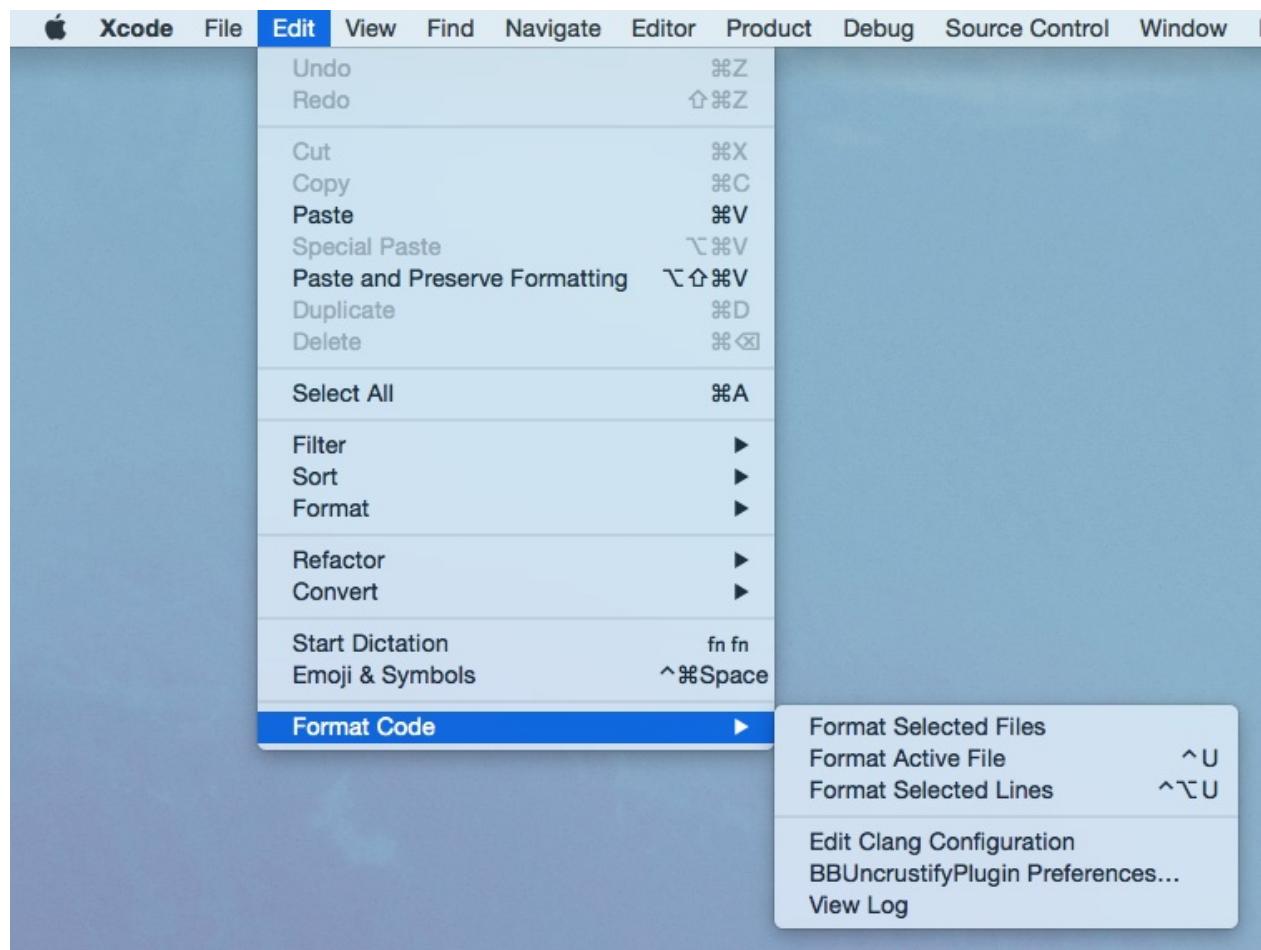
## 参考链接

1. [Xcode插件 BBUncrustifyPlugin-Xcode](#)
2. [Uncrustify/UncrustifyX 配置工具](#)
3. [GitHub - BBUncrustifyPlugin-Xcode](#)
4. [GitHub - UncrustifyX](#)

## 详细内容

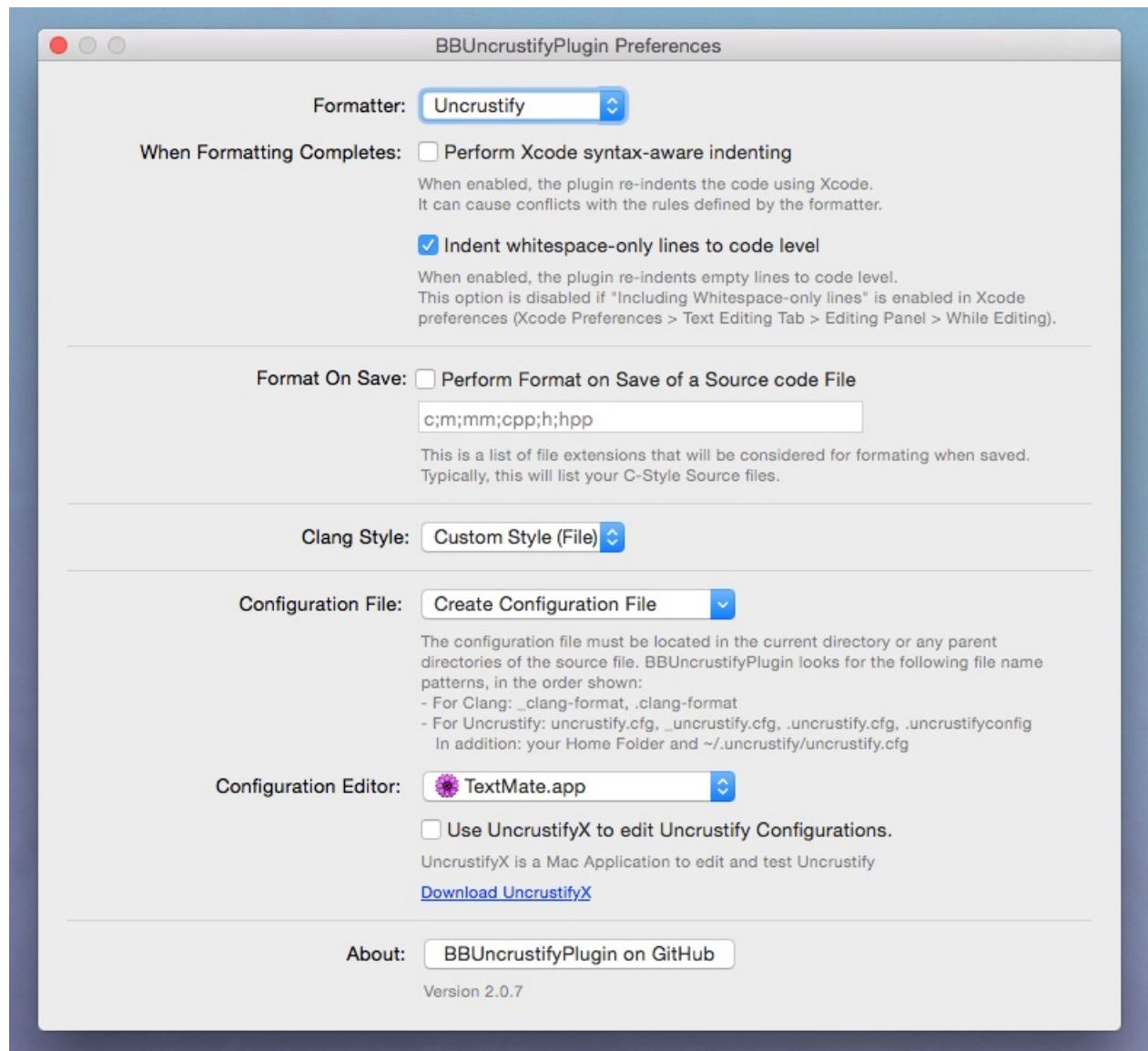
### 1. 安装（仅介绍最简单方式，更多安装方式参考[GitHub](#)）

1. [点击这里](#) 下载 BBUncrustifyPlugin-Xcode 最新发布版本
2. 解压下载文件，复制 *UncrustifyPlugin.xcplugin* 到 *~/Library/Application Support/Developer/Shared/Xcode/Plug-ins* 文件夹下
3. 重启 Xcode，即可在 *Edit* 选项下看到 *Format Code*



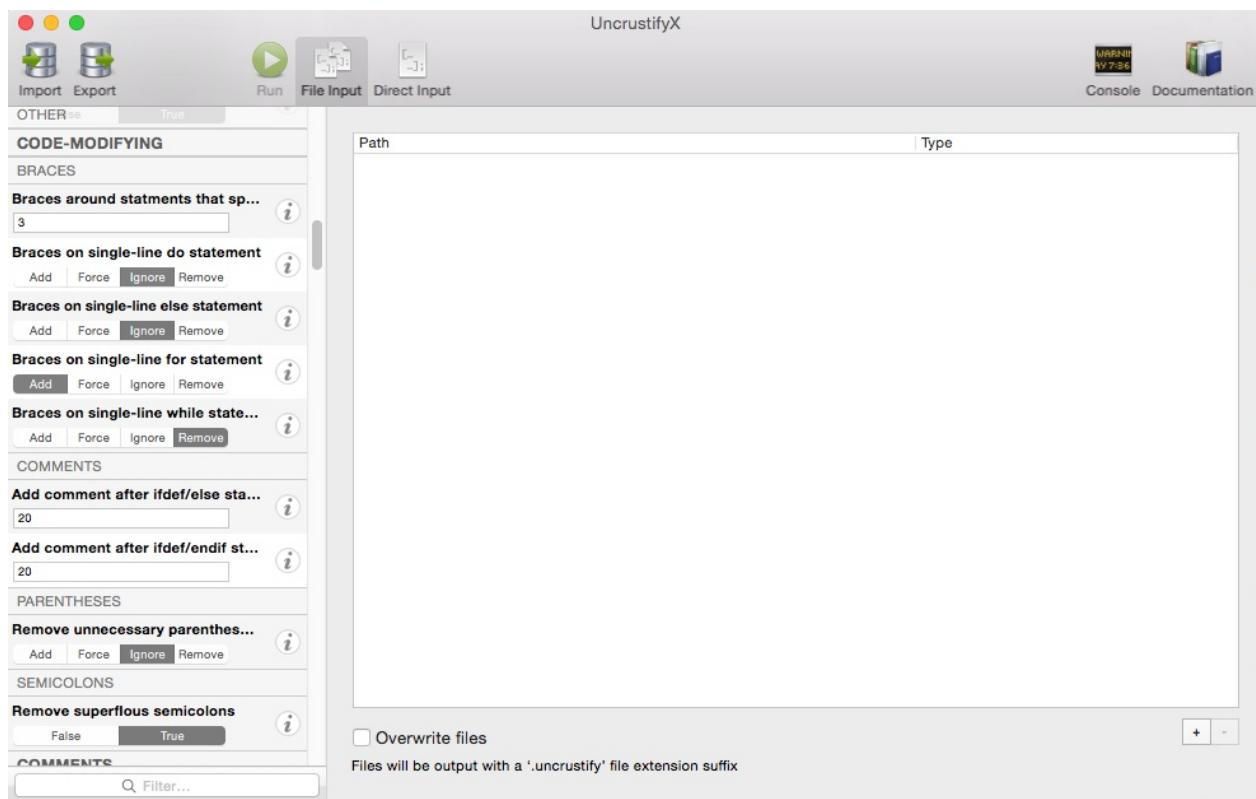
## 2. 使用（仅介绍快速使用方法，使用[ryan-objc.cfg](#)作为初始 **Objective-C** 配置文件）

1. 点击[这里](#) 下载 Objective-C 配置文件
2. 复制 `ryan-objc.cfg` 到 `~/.uncrustify` 文件夹下（若没有则创建 `.uncrustify` 文件夹）
3. 进入 `Xcode`，选中需要格式化的代码，选择 `Edit -> Format Code -> Format Selected Lines` 即可格式化代码



### 3. 配置 (UncrustifyX 配置工具的下载使用)

1. 点击这里 下载 UncrustifyX 最新发布版本
2. 解压下载文件，拖动 UncrustifyX 至 Applications 文件夹
3. 进入 Xcode，选择 Edit -> Format Code -> BBUncrustifyPlugin Preferences，设置 Configuration Editor 为 UncrustifyX
4. 选择 Edit -> Format Code -> Edit Uncrustify Configuration 即可通过 UncrustifyX 配置 cfg 文件



## 效果图

(无)

## 备注

ryan-objc.cfg 文件

```
#  
# Uncrustify Configuration File  
# File Created With Uncrustify 0.2 (140)  
#  
  
# Alignment  
# -----  
  
## Alignment  
  
# Align obj-c declaration params on colon  
align_oc_decl_colon = true # boolean  
n (false/true)
```

```
# Align on tabstop
align_on_tabstop = false      # boolean
n (false/true)

# Align variable definitions
align_func_params = true      # boolean
n (false/true)

# Align with tabs
align_with_tabs = false       # boolean
n (false/true)

# Keep non-indenting tabs
align_keep_tabs = false       # boolean
n (false/true)

## Alignment Span

# Alignment span for #define bodies
align_pp_define_span = 8      # number

# Alignment span for equals in enums
align_enum_equ_span = 4        # number

# Alignment span for obj-c message colons
align_oc_msg_colon_span = 20   # number

# Alignment span for obj-c message spec
align_oc_msg_spec_span = 20   # number

# Alignment span for single-line typedefs
align_typedef_span = 5         # number

# Alignment span for struct initializer values
align_struct_init_span = 4     # number

# Alignment span for trailing comments
align_right_cmt_span = 8       # number
```

```
## Alignment Style

# Alignment style for star in variable definitions
align_var_def_star_style = 1 # number

## Gap

# Minimum gap between type and synonym of typedef
align_TYPEDEF_gap = 3 # number

# Minimum gap for trailing comment
align_right_cmt_gap = 8 # number

## Other

# Always align with first parameter in obj-c message
align_oc_msg_colon_first = false # boolean (false/true)

# Blank Lines
# -------

## Newline Count After

# Newline count after function body
nl_after_func_body = 2 # number

# Newline count after single-line function body
nl_after_func_body_one_liner = 2 # number

# Newline count after variable definition block
nl_func_var_def_blk = 1 # number

## Other

# Remove blank lines after open brace
eat_blanks_after_open_brace = true # boolean (false/true)

# Remove blank lines before close brace
```

```
eat_blanks_before_close_brace          = true           # boolean
n (false/true)

# Code-Modifying
# -------

## Braces

# Braces around statements that span N newlines
mod_full_brace_nl                   = 3              # number

# Braces on single-line do statement
mod_full_brace_do                  = ignore         # string
(add/force/ignore/remove)

# Braces on single-line else statement
mod_full_brace_if                  = ignore         # string
(add/force/ignore/remove)

# Braces on single-line for statement
mod_full_brace_for                = add            # string
(add/force/ignore/remove)

# Braces on single-line while statement
mod_full_brace_while              = remove         # string
(add/force/ignore/remove)

## Comments

# Add comment after ifdef/else statement of size
mod_add_long_ifdef_else_comment    = 20             # number

# Add comment after ifdef/endif statement of size
mod_add_long_ifdef_endif_comment   = 20             # number

## Parentheses

# Remove unnecessary parentheses on return statement
mod_paren_on_return               = ignore         # string
(add/force/ignore/remove)
```

```
## Semicolons

# Remove superflous semicolons
mod_remove_extra_semicolon          = true           # boolean
n (false/true)

# Comments
# -------

## Empty Lines

# Empty first line for multi-line C comments
cmt_c_nl_start                      = true           # boolean
n (false/true)

# Empty first line for multi-line C++ comments
cmt_cpp_nl_start                     = true           # boolean
n (false/true)

## Other

# Stars on multi-line comments
cmt_star_cont                        = false          # boolean
n (false/true)

# General
# -------

## Other

# Newline character
newlines                             = lf            # string
(auto/cr/crlf/lf)

# Output tab size
output_tab_size                       = 4             # number

# Indentation
# -----
```

```
## Indentation

# Indent obj-c block
indent_oc_block = true # boolean
n (false/true)

## Indentation Size

# Indentation column size
indent_columns = 4 # number

# Indentation size between case and switch
indent_switch_case = 4 # number

# Indentation size for obj-c blocks in a message parameter
indent_oc_block_msg = 4 # number

# Indentation size for obj-c message subsequent parameters
indent_oc_msg_colon = 4 # number

## Other

# Align continued statements at equals
indent_align_assign = false # boolean
n (false/true)

# Indent goto labels
indent_label = 2 # number

# Indent with tabs
indent_with_tabs = 0 # number

# Newlines
# -------

## Merging

# Change unbraced if statements into one-liner
nl_create_if_one_liner = true # boolean
```

```
n (false/true)

## Newline After

# Newline after brace open
nl_after_brace_open = false      # boolean
n (false/true)

# Newline after for
nl_after_for = force            # string
(add/force/ignore/remove)

# Newline after if
nl_after_if = force             # string
(add/force/ignore/remove)

# Newline after macro multi-line definition
nl_multi_line_define = true     # boolean
n (false/true)

# Newline after return
nl_after_return = true           # boolean
n (false/true)

## Newline Before

# Newline before case statement
nl_before_case = true            # boolean
n (false/true)

# Newline before for
nl_before_for = force            # string
(add/force/ignore/remove)

# Newline before if
nl_before_if = force             # string
(add/force/ignore/remove)

# Newline before while
nl_before_while = force          # string
```

```
(add/force/ignore/remove)

## Newline Between

# Newline between case colon and open brace
nl_case_colon_brace          = remove      # string
(add/force/ignore/remove)

# Newline between catch and open brace
nl_catch_brace                = remove      # string
(add/force/ignore/remove)

# Newline between close brace and catch
nl_braceCatch                 = remove      # string
(add/force/ignore/remove)

# Newline between close brace and else
nl_braceElse                  = remove      # string
(add/force/ignore/remove)

# Newline between close brace and finally
nl_braceFinally               = remove      # string
(add/force/ignore/remove)

# Newline between close brace and while
nl_braceWhile                 = remove      # string
(add/force/ignore/remove)

# Newline between close parenthesis and open brace in multi line
# conditional
nl_multi_line_cond             = false       # boolean
n (false/true)

# Newline between do and open brace
nl_do_brace                   = remove      # string
(add/force/ignore/remove)

# Newline between else and open brace
nl_else_brace                 = remove      # string
(add/force/ignore/remove)
```

```
# Newline between else if and open brace
nl_elseif_brace = remove # string
(add/force/ignore/remove)

# Newline between enum and open brace
nl_enum_brace = remove # string
(add/force/ignore/remove)

# Newline between finally and open brace
nl_finally_brace = remove # string
(add/force/ignore/remove)

# Newline between for and open brace
nl_for_brace = remove # string
(add/force/ignore/remove)

# Newline between function call and open brace
nl_fcall_brace = remove # string
(add/force/ignore/remove)

# Newline between function signature and open brace
nl_fdef_brace = remove # string
(add/force/ignore/remove)

# Newline between if and open brace
nl_if_brace = remove # string
(add/force/ignore/remove)

# Newline between struct and open brace
nl_struct_brace = remove # string
(add/force/ignore/remove)

# Newline between switch and open brace
nl_switch_brace = remove # string
(add/force/ignore/remove)

# Newline between try and open brace
nl_try_brace = remove # string
(add/force/ignore/remove)
```

```
# Newline between union and open brace
nl_union_brace = remove # string
(add/force/ignore/remove)

# Newline between while and open brace
nl_while_brace = remove # string
(add/force/ignore/remove)

## Other

# Don't split one-line obj-c messages
nl_oc_msg_leave_one_liner = true # boolean
n (false/true)

# Newlines at end of file
nl_end_of_file = add # string
(add/force/ignore/remove)

# Place obj-c message parameters on new lines
nl_oc_msg_args = true # boolean
n (false/true)

# Spacing
# -----

## Space After

# Space after C++ comment opening
sp_cmt_cpp_start = ignore # string
(add/force/ignore/remove)

# Space after cast
sp_after_cast = remove # string
(add/force/ignore/remove)

# Space after class colon
sp_after_class_colon = force # string
(add/force/ignore/remove)
```

```
# Space after comma
sp_after_comma = add # string
(add/force/ignore/remove)

# Space after condition close parenthesis
sp_after_sparen = force # string
(add/force/ignore/remove)

# Space after obj-c block caret
sp_after_oc_block_caret = remove # string
(add/force/ignore/remove)

# Space after obj-c colon
sp_after_oc_colon = remove # string
(add/force/ignore/remove)

# Space after obj-c dictionary colon
sp_after_oc_dict_colon = add # string
(add/force/ignore/remove)

# Space after obj-c message colon
sp_after_send_oc_colon = remove # string
(add/force/ignore/remove)

# Space after obj-c property
sp_after_oc_property = add # string
(add/force/ignore/remove)

# Space after obj-c return type
sp_after_oc_return_type = remove # string
(add/force/ignore/remove)

# Space after obj-c scope
sp_after_oc_scope = force # string
(add/force/ignore/remove)

# Space after obj-c type
sp_after_oc_type = remove # string
(add/force/ignore/remove)
```

```
# Space after pointer star
sp_after_ptr_star = remove # string
(add/force/ignore/remove)

# Space after pointer star followed by function
sp_after_ptr_star_func = force # string
(add/force/ignore/remove)

## Space Around

# Space around arithmetic operators
sp_arith = add # string
(add/force/ignore/remove)

# Space around assignment operator
sp_assign = add # string
(add/force/ignore/remove)

# Space around boolean operators
sp_bool = add # string
(add/force/ignore/remove)

# Space around compare operators
sp_compare = add # string
(add/force/ignore/remove)

# Space around ternary condition colon
sp_cond_colon = force # string
(add/force/ignore/remove)

# Space around ternary condition question mark
sp_cond_question = force # string
(add/force/ignore/remove)

## Space Before

# Space before case colon
sp_before_case_colon = remove # string
(add/force/ignore/remove)
```

```
# Space before class colon
sp_before_class_colon = force # string
(add/force/ignore/remove)

# Space before if/for/switch/while open parenthesis
sp_before_sparen = force # string
(add/force/ignore/remove)

# Space before obj-c block caret
sp_before_oc_block_caret = ignore # string
(add/force/ignore/remove)

# Space before obj-c colon
sp_before_oc_colon = remove # string
(add/force/ignore/remove)

# Space before obj-c dictionary colon
sp_before_oc_dict_colon = remove # string
(add/force/ignore/remove)

# Space before obj-c message colon
sp_before_send_oc_colon = remove # string
(add/force/ignore/remove)

# Space before pointer star
sp_before_ptr_star = force # string
(add/force/ignore/remove)

# Space before pointer star followed by function
sp_before_ptr_star_func = force # string
(add/force/ignore/remove)

# Space before unnamed pointer star
sp_before_unnamed_ptr_star = ignore # string
(add/force/ignore/remove)

## Space Between

# Space between @selector and open parenthesis
sp_after_oc_at_sel = remove # string
```

```
(add/force/ignore/remove)

# Space between catch and open brace
sp_catch_brace = add # string
(add/force/ignore/remove)

# Space between catch and open parenthesis
sp_catch_paren = add # string
(add/force/ignore/remove)

# Space between close brace and else
sp_brace_else = force # string
(add/force/ignore/remove)

# Space between close parenthesis and open brace
sp_paren_brace = force # string
(add/force/ignore/remove)

# Space between closing brace and catch
sp_braceCatch = add # string
(add/force/ignore/remove)

# Space between closing brace and finally
sp_brace_finally = add # string
(add/force/ignore/remove)

# Space between closing parenthesis and open brace
sp_fparen_brace = force # string
(add/force/ignore/remove)

# Space between else and open brace
sp_else_brace = force # string
(add/force/ignore/remove)

# Space between finally and open brace
sp_finally_brace = add # string
(add/force/ignore/remove)

# Space between function name and open parenthesis
sp_func_call_paren = remove # string
```

```
(add/force/ignore/remove)

# Space between function name and open parenthesis in declaration
sp_func_proto_paren = remove      # string
(add/force/ignore/remove)

# Space between function name and open parenthesis in function definition
sp_func_def_paren = remove      # string
(add/force/ignore/remove)

# Space between pointer stars
sp_between_ptr_star = remove      # string
(add/force/ignore/remove)

# Space between sizeof and open parenthesis
sp_sizeof_paren = remove      # string
(add/force/ignore/remove)

# Space between try and open brace
sp_try_brace = add      # string
(add/force/ignore/remove)

## Space Inside

# Space inside @selector() parens
sp_inside_oc_at_sel_parens = remove      # string
(add/force/ignore/remove)

# Space inside braces
sp_inside_braces = add      # string
(add/force/ignore/remove)

# Space inside cast parentheses
sp_inside_paren_cast = remove      # string
(add/force/ignore/remove)

# Space inside enum braces
sp_inside_braces_enum = add      # string
```

```
(add/force/ignore/remove)

# Space inside function parentheses
sp_inside_fparen = remove      # string
(add/force/ignore/remove)

# Space inside if-condition parentheses
sp_inside_sparen = remove      # string
(add/force/ignore/remove)

# Space inside parentheses
sp_inside_paren = remove       # string
(add/force/ignore/remove)

# Space inside parentheses in function type
sp_inside_tparen = remove      # string
(add/force/ignore/remove)

# Space inside struct/union braces
sp_inside_braces_struct = add   # string
(add/force/ignore/remove)
```

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-08 | Alfred Jiang | -  |

## 方案名称

Xcode - 使用 chisel 提高 LLDB 调试效率

## 关键字

Xcode \ 调试 \ 断点 \ 页面 \ 约束

## 需求场景

1. 需要对复杂页面进行调试时

## 参考链接

1. GitHub - chisel
2. 简书 - LLdb篇2教你使用faceBook的chisel来提高调试效率
3. 简书 - 小笨狼的LLDB技巧:chisel

## 详细内容

### 安装

1. 安装 **Homebrew** (如果已经安装了，可以跳过这一步)

```
ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

2. 更新 **Homebrew** (如果已经最新了，可以跳过这一步)

```
brew update
```

### 3. 通过 Homebrew 安装 chisel

```
brew install chisel
```

### 4. 添加 chisel 命令至 ~/.lldbinit

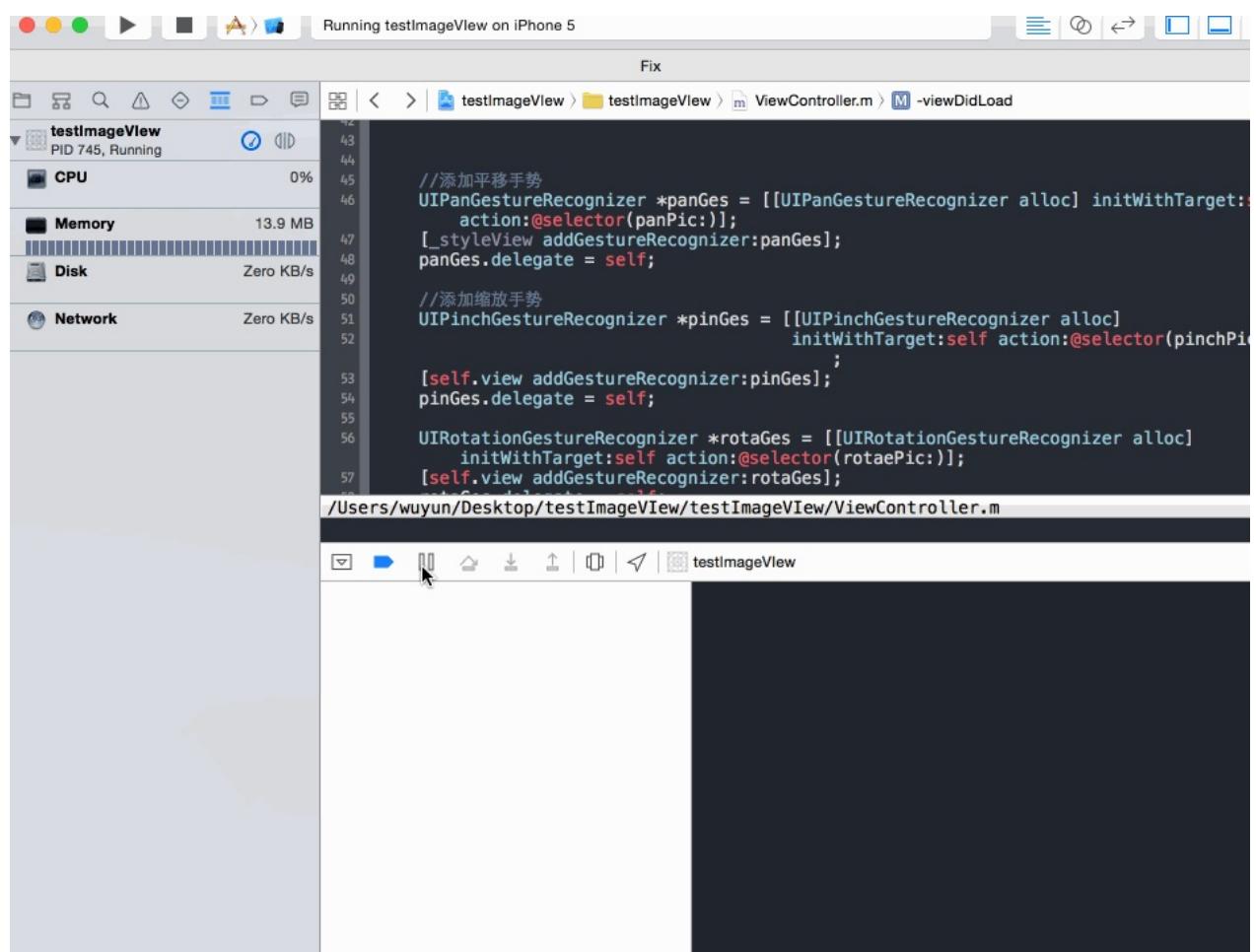
```
touch ~/.lldbinit echo "command script import /usr/local/opt/chisel/libexec/fbllldb.py" >> ~/.lldbinit
```

### 5. 更新 chisel

```
brew upgrade chisel
```

## 使用

### 1. 通过 pviews 命令直接查看页面层级关系



### 2. 通过 border & unborder 命令为页面控件添加和删除边框颜色和边框的宽度

```
(lldb) border 0x79ec3140 -c green -w 2
```

3. 通过 **paltrace** 命令查看是否存在 **Ambiguous Layouts** (**AMBIGUOUS LAYOUT** 即为约束冲突或不完善位置)，该命令相当于对 **View** 调用 **\_autolayoutTrace**

```
(lldb) paltrace
• UIWindow:0x7ff450d2fb50 - AMBIGUOUS LAYOUT
|   • UIView:0x7ff450e14430
|   |   *_UILayoutGuide:0x7ff450d30e90
|   |   *_UILayoutGuide:0x7ff450d31230
|   |   *UIView:0x7ff450d32870- AMBIGUOUS LAYOUT for UIView:0x7ff450d32870.minX{id: 46}
```

Legend:

- \* - is laid out with auto layout
- + - is laid out manually, but is represented in the layout engine because translatesAutoresizingMaskIntoConstraints = YES
- - layout engine host

4. 通过 **alamborder & alamunborder** 命令为 **Ambiguous Layouts** (**AMBIGUOUS LAYOUT** 即为约束冲突或不完善位置) 控件添加和删除边框颜色和边框的宽度

```
(lldb) alamborder
```

5. 通过 **pinternals** 命令打印对象成员变量(**model** 为某实例对象)

```
(lldb) pinternals model
```

6. 使用 **pkp** 命令打印属性，该命令是 **po** 命令的升级版，**po obj.xxx** 是调用 **getter** 方法，如果没有 **getter** 方法就无法打印了。**pkp obj.xxx** 优先调用 **getter** 方法，没有 **getter** 方法会直接查找成员变量。

```
(lldb) pkp self.view
```

7. 通过 **flicker** 命令查看 **View**, 将 **View** 闪烁一下，以便于查找 **View** 的位置

```
(lldb) flicker self.subView
```



8. 通过 **caflush** 命令更新 **UI**, 该命令主要用于通过 **LLDB** 修改界面控件显示属性之后刷新页面

```
caflush
```

9 通过 **show & hide** 命令显示和隐藏页面显示控件

```
(lldb) show 0x7fe713901f10
```

```
(lldb) hide 0x7fe713901f10
```

10. 通过 **visualize** 命令打开mac下的预览app打开我们的图片 **UIImage**, **CGImageRef** 格式的图片，甚至 **view** 和 **layer** 的图片

```
(lldb) visualize image
```

更多命令使用请关注 参考链接

效果图

(无)

备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-01 | Alfred Jiang | -  |

## 方案名称

Xcode - 使用 DWUReturningAlert 进行 UITableView 和 UICollectionView 的绘图性能测试

## 关键字

Xcode \ DWUReturningAlert \ UITableView \ UICollectionView \ 绘图性能测试

## 需求场景

1. 需要对 UITableView 和 UICollectionView 的绘图进行性能测试时

## 参考链接

1. [GitHub - DWUReturningAlert](#)

## 详细内容

### 使用方法

在工程中添加 [DWUReturningAlert.m](#) 文件即可

### Objective-C

```
//DWUReturningAlert.m
//Copyright (c) 2015 Di Wu
//
//Permission is hereby granted, free of charge, to any person obtaining a copy
```

```
//of this software and associated documentation files (the "Software"), to deal
//in the Software without restriction, including without limitation the rights
//to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
//copies of the Software, and to permit persons to whom the Software is
//furnished to do so, subject to the following conditions:
//
//The above copyright notice and this permission notice shall be included in
//all copies or substantial portions of the Software.
//
//THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND
//, EXPRESS OR
//IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
//FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
//AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
//OR OTHER
//LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE
//, ARISING FROM,
//OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
//DEALINGS IN
//THE SOFTWARE.

// Comment out if you want to disable this entire runtime hack
#define DWUReturningAlertEnabled

#if defined (DEBUG) && defined (DWUReturningAlertEnabled)

#import <Foundation/Foundation.h>
#import <objc/runtime.h>
#import <objc/message.h>
#import <UIKit/UITableViewCell.h>
#import <UIKit/UIImage.h>
#import <UIKit/UITableView.h>
#import <UIKit/UILabel.h>
```

```
#import <QuartzCore/CALayer.h>
#import <UIKit/UINibLoading.h>
#import <UIKit/UICollectionViewCell.h>
#import <UIKit/UICollectionView.h>
#import <UIKit/UITableViewHeaderFooterView.h>

// ----- UI Configuration -----
static const CGFloat DWU_BORDER_WIDTH = 5.0;

static const CGFloat DWU_LABEL_HEIGHT = 16.0;

static const CGFloat DWU_LABEL_WIDTH_UITABLEVIEW_CELL = 240.0;

static const CGFloat DWU_LABEL_WIDTH_UICOLLECTIONVIEW_CELL = 50.0
;

static const CGFloat DWU_LABEL_FONT_SIZE = 12.0;

static NSString *DWU_LABEL_FORMAT_UITABLEVIEW_CELL = @"cellForRo
w: %zd ms, drawRect: %zd ms";

static NSString *DWU_LABEL_FORMAT_UITABLEVIEW_HEADER = @"viewFor
Header: %zd ms, drawRect: %zd ms";

static NSString *DWU_LABEL_FORMAT_UITABLEVIEW_FOOTER = @"viewFor
Footer: %zd ms, drawRect: %zd ms";

static NSString *DWU_LABEL_FORMAT_UICOLLECTIONVIEW_CELL = @"%zd
/ %zd";

#define DWU_BORDER_COLOR [[UIColor redColor] CGColor]

#define DWU_TEXT_LABEL_BACKGROUND_COLOR [UIColor blackColor]

#define DWU_TEXT_LABEL_FONT_COLOR [UIColor whiteColor]
// -----


static const NSInteger DWU_TIME_INTERVAL_LABEL_TAG = NSIntegerMax
- 123;
```

```
static char DWU_CALAYER_ASSOCIATED_OBJECT_KEY;

static char DWU_UIVIEW_TABLEVIEW_CELL_DELEGATE_ASSOCIATED_OBJECT
_KEY;

static char DWU_UIVIEW_DRAW_RECT_TIME_COUNT_NUMBER_ASSOCIATED_OB
JECT_KEY;

static char DWU_UIVIEW_CELL_FOR_ROW_TIME_COUNT_NUMBER_ASSOCIATED
_OBJECT_KEY;

typedef id(^CellForRowAtIndexPathBlock)(__unsafe_unretained id _self,
                                         __unsafe_unretained id arg1,
                                         __unsafe_unretained id arg2);

typedef id(^CollectionHeaderFooterBlock)(__unsafe_unretained id _self,
                                         __unsafe_unretained id arg1,
                                         __unsafe_unretained id arg2,
                                         __unsafe_unretained id arg3);

#pragma mark - swizzling method from block

// http://www.mikeash.com/pyblog/friday-qa-2010-01-29-method-replacement-for-fun-and-profit.html
static BOOL dwu_replaceMethodWithBlock(Class c, SEL origSEL, SEL newSEL, id block) {
    if ([c instancesRespondToSelector:newSEL]) {
        return YES;
    }
    Method origMethod = class_getInstanceMethod(c, origSEL);
    IMP impl = imp_implementationWithBlock(block);
    if (!class_addMethod(c, newSEL, impl, method_getTypeEncoding(origMethod))) {
        return NO;
    }else {
        Method newMethod = class_getInstanceMethod(c, newSEL);
        if (class_addMethod(c, origSEL, method_getImplementation(newMethod),
                           method_getTypeEncoding(origMethod))) {
            class_replaceMethod(c, newSEL, method_getImplementation(origMethod),
                               method_getTypeEncoding(newMethod));
        }else {
            method_exchangeImplementations(origMethod, newMethod);
        }
    }
}
```

```
);

    }

}

return YES;
}

#pragma mark - time count label

@interface DWUKVOLabel : UILabel

// Known issue: *strong* will lead to retain cycle.
// (While *weak* will lead to a NSKVODeallocate exception.)
// Will adopt something like FBKVOController in the future.
@property (nonatomic, strong) UIView *observedView;

@property (nonatomic, assign) NSInteger cellForRowTimeInteger;

@property (nonatomic, assign) NSInteger drawRectTimeInteger;

@property (nonatomic, copy) NSString *format;

- (instancetype)initWithKVOTarget: (UIView *)view frame: (CGRect)frame;

@end

@implementation DWUKVOLabel

- (instancetype)initWithKVOTarget: (UIView *)view frame: (CGRect)frame {
    if ((self = [super initWithFrame:frame])) {
        _observedView = view;
        _cellForRowTimeInteger = 0;
        _drawRectTimeInteger = 0;
        [view addObserver:self forKeyPath:@"dwuCellForRowTimeCountNumber" options:NSKeyValueObservingOptionNew | NSKeyValueObservingOptionInitial context:nil];
        [view addObserver:self forKeyPath:@"dwuDrawRectTimeCountNumber" options:NSKeyValueObservingOptionNew | NSKeyValueObservingOptionInitial context:nil];
    }
}
```

```
    }

    return self;
}

- (void)observeValueForKeyPath:(NSString *)keyPath
                      ofObject:(id)object
                        change:(NSDictionary *)change
                       context:(void *)context
{
    NSNumber *number = [change objectForKey:NSKeyValueChangeNewKey];
    if (!number || ![number isKindOfClass:[NSNumber class]]) {
        return;
    }

    if ([keyPath isEqualToString:@"dwuCellForRowTimeCountNumber"])
    {
        self.cellForRowTimeInteger = [number integerValue];
    } else if ([keyPath isEqualToString:@"dwuDrawRectTimeCountNumber"])
    {
        self.drawRectTimeInteger += [number integerValue];
    }

    [self updateText];
}

- (void)updateText {
    self.text = [NSString stringWithFormat:self.format, self.cellForRowTimeInteger, self.drawRectTimeInteger];
}

- (void)dealloc {
    [self.observedView removeObserver:self forKeyPath:@"dwuCellForRowTimeCountNumber"];
    [self.observedView removeObserver:self forKeyPath:@"dwuDrawRectTimeCountNumber"];
}

@end
```

```
#pragma mark - Category

@interface UIView (DWURecyclingAlert)

@property (nonatomic, unsafe_unretained) UIView *dwuCellDelegate;

@property (nonatomic, strong) NSNumber *dwuDrawRectTimeCountNumber;

@property (nonatomic, strong) NSNumber *dwuCellForRowTimeCountNumber;

@end

@implementation UIView (DWURecyclingAlert)

- (void)setDwuCellDelegate:(UIView *)delegate {
    objc_setAssociatedObject(self, &DWU_UVIEW_TABLEVIEW_CELL_DELEGATE_ASSOCIATED_OBJECT_KEY, delegate, OBJC_ASSOCIATION_ASSIGN);
}

- (UIView *)dwuCellDelegate {
    UITableViewCell *delegate = objc_getAssociatedObject(self, &DWU_UVIEW_TABLEVIEW_CELL_DELEGATE_ASSOCIATED_OBJECT_KEY);
    return delegate;
}

- (void)setDwuDrawRectTimeCountNumber: (NSNumber *)number {
    objc_setAssociatedObject(self, &DWU_UVIEW_DRAW_RECT_TIME_COUNT_NUMBER_ASSOCIATED_OBJECT_KEY, number, OBJC_ASSOCIATION_RETAIN_NONATOMIC);
}

- (NSNumber *)dwuDrawRectTimeCountNumber {
    NSNumber *number = objc_getAssociatedObject(self, &DWU_UVIEW_DRAW_RECT_TIME_COUNT_NUMBER_ASSOCIATED_OBJECT_KEY);
    return number;
}
```

```
- (void)setDwuCellForRowTimeCountNumber: (NSNumber *)number {
    objc_setAssociatedObject(self, &DWU_UVIEW_CELL_FOR_ROW_TIME
    _COUNT_NUMBER_ASSOCIATED_OBJECT_KEY, number, OBJC_ASSOCIATION_RETAIN_NONATOMIC);
}

- (NSNumber *)dwuCellForRowTimeCountNumber {
    NSNumber *number = objc_getAssociatedObject(self, &DWU_UVIEW_CELL_FOR_ROW_TIME_COUNT_NUMBER_ASSOCIATED_OBJECT_KEY);
    return number;
}

@end

@interface CALayer (DWUReturningAlert)

@property (nonatomic, assign) NSInteger dwuRecyclingCount;

@end

@implementation CALayer (DWUReturningAlert)

- (void)setDwuRecyclingCount:(NSInteger)recyclingCount {
    objc_setAssociatedObject(self, &DWU_CALAYER_ASSOCIATED_OBJECT_KEY, @(recyclingCount), OBJC_ASSOCIATION_RETAIN_NONATOMIC);
}

- (NSInteger)dwuRecyclingCount {
    NSNumber *recyclingCountNumber = objc_getAssociatedObject(self, &DWU_CALAYER_ASSOCIATED_OBJECT_KEY);
    return [recyclingCountNumber integerValue];
}

- (void)dwu_addRedBorderEffect {
    self.borderColor = DWU_BORDER_COLOR;
    self.borderWidth = DWU_BORDER_WIDTH;
}

- (void)dwu_removeRedBorderEffect {
```

```
    self.borderColor = [[UIColor clearColor] CGColor];
    self.borderWidth = 0.0;
}

static BOOL dwu_implementsSelector(id obj, SEL sel) {
    if ([[obj class] instanceMethodForSelector:sel] != [[obj superclass] instanceMethodForSelector:sel]) {
        return YES;
    } else {
        return NO;
    }
}

static void dwu_swizzleDrawRectIfNotYet(CALayer *layer) {
    if (!layer.delegate) {
        return;
    }
    if (![layer.delegate isKindOfClass:[UIView class]]) {
        return;
    }
    UIView *containerView = layer.delegate;
    if (!dwu_implementsSelector(containerView, @selector(drawRect:))) {
        return;
    }
    Class c = containerView.class;
    if ([NSStringFromClass(c) hasPrefix:@"UI"]) {
        return;
    }
    static NSMutableSet *classSet;
    if (!classSet) {
        classSet = [NSMutableSet set];
    }
    if ([classSet containsObject:c]) {
        return;
    }
    [classSet addObject:c];
    SEL selector = @selector(drawRect:);
    NSString *selStr = NSStringFromSelector(selector);
    SEL newSelector = NSSelectorFromString([NSString stringWithFormat:@"->%s", selStr]);
    if (objc_setAssociatedObject(containerView, &DWU_DRAW_RECT_SELECTOR, newSelector, OBJC_ASSOCIATION_COPY_NONRESERVED)) {
        return;
    }
    dwu_swizzleDrawRectIfNotYet(containerView.superlayer);
}
```

```
ormat:@"dwu_%@", selStr]);
    dwu_replaceMethodWithBlock(c, selector, newSelector, ^(__unsafe_unretained UIView *containerView, CGRect rect) {
        NSDate *date = [NSDate date];
        containerView.opaque = NO;
        ((void (*) (id, SEL, CGRect))objc_msgSend)(containerView, newSelector, rect);
        NSTimeInterval timeInterval = ceilf(-[date timeIntervalSinceNow] * 1000);
        containerView.dwuCellDelegate.dwuDrawRectTimeCountNumber = @(timeInterval);
    });
}

- (void)dwu_scanLayerHierarchyRecursively {
    dwu_swizzleDrawRectIfNotYet(self);
    static NSMapTable *cgImageRefDict;
    if (!cgImageRefDict) {
        cgImageRefDict = [NSMapTable mapTableWithKeyOptions:NSMapTableCopyIn
                                                       valueOptions:NSMapTableWeakMemory];
    }
    NSInteger recyclingCount = self.dwuRecyclingCount;
    SEL imageSelector = @selector(image);
    BOOL viewTargetFound = NO;
    BOOL imageTargetFound = NO;
    if (self.delegate && [self.delegate respondsToSelector:@selector(imageForView:)]) {
        UIImage *image = ((UIImage * (^)(id, SEL))objc_msgSend)(self.delegate, imageSelector);
        if (image) {
            NSString *addressString = [NSString stringWithFormat:@"%@", image.CGImage];
            if (![cgImageRefDict objectForKey:addressString]) {
                [cgImageRefDict setObject:self.delegate forKey:addressString];
                imageTargetFound = YES;
            } else {
                UIView *someLastMarkedView = [cgImageRefDict obj
```

```
ectForKey:addressString];
    [someLastMarkedView.layer dwu_removeRedBorderEffect];
}
}
} else if (!recyclingCount && self.superlayer && self.superlayer.dwuRecyclingCount) {
    viewTargetFound = YES;
}

if (viewTargetFound || imageTargetFound) {
    [self dwu_addRedBorderEffect];
} else {
    [self dwu_removeRedBorderEffect];
}
UIView *cellDelegate = [self dwu_findCell];
[self dwu_injectLayer:cellDelegate.layer withCellDelegate:cellDelegate];
for (CALayer *sublayer in self.sublayers) {
    [self dwu_injectLayer:sublayer withCellDelegate:cellDelegate];
    [sublayer dwu_scanLayerHierarchyRecursively];
}
self.dwuRecyclingCount++;
}

- (UIView *)dwu_findCell {
    UIView *containerView = self.delegate;
    if (!containerView) {
        return nil;
    }
    if (![containerView isKindOfClass:[UIView class]]) {
        return nil;
    }
    if (containerView.dwuCellDelegate) {
        return containerView.dwuCellDelegate;
    } else if ([containerView isKindOfClass:[UITableViewCell class]]) {
        return containerView;
    } else if ([containerView isKindOfClass:[UITableViewHeaderFo
```

```
oterView class]]) {
    return containerView;
} else if ([containerView isKindOfClass:[UICollectionViewReusabl
eView class]]) {
    return containerView;
} else {
    return nil;
}
}

- (void)dwu_injectLayer: (CALayer *)layer withCellDelegate:(UIVi
ew *)cellDelegate {
    if (layer.delegate && [layer.delegate isKindOfClass:[UIView
class]]) {
        UIView *containerView = layer.delegate;
        containerView.dwuCellDelegate = cellDelegate;
    }
}

@end

#pragma mark - generate for UITableViewCell / UICollectionViewCe
ll labels

static CellForRowAtIndexPathBlock dwu_generateTimeLabel(SEL targ
etSelector, CGFloat labelWidth, NSString *timeStringFormat) {
    return ^(__unsafe_unretained UITableView *_self, __unsafe_un
retained id arg1, __unsafe_unretained id arg2) {
        NSDate *date = [NSDate date];
        UIView *returnView = ((UIView * (^)(id, SEL, id, id))ob
jc_msgSend)(_self, targetSelector, arg1, arg2);
        NSTimeInterval timeInterval = ceilf(-[date timeIntervals
inceNow] * 1000);
        [[returnView layer] dwu_scanLayerHierarchyRecursively];
        DWUKVOLabel *timeIntervalLabel = (DWUKVOLabel *)[returnV
iew viewWithTag:DWU_TIME_INTERVAL_LABEL_TAG];
        if (!timeIntervalLabel) {
            timeIntervalLabel = [[DWUKVOLabel alloc] initWithKVO
Target:returnView frame:CGRectMake(0, 0, labelWidth, DWU_LABEL_H
EIGHT)];
        }
    }
}
```

```
        timeIntervalLabel.userInteractionEnabled = NO;
        timeIntervalLabel.backgroundColor = DWU_TEXT_LABEL_B
ACKGROUND_COLOR;
        timeIntervalLabel.textColor = DWU_TEXT_LABEL_FONT_CO
LOR;
        timeIntervalLabel.font = [UIFont boldSystemFontOfSize:DWU_LABEL_FONT_SIZE];
        timeIntervalLabel.textAlignment = NSTextAlignmentCenter;
        timeIntervalLabel.adjustsFontSizeToFitWidth = YES;
        timeIntervalLabel.tag = DWU_TIME_INTERVAL_LABEL_TAG;
        timeIntervalLabel.layer.dwuRecyclingCount++;
        [returnView addSubview:timeIntervalLabel];
    }
    timeIntervalLabel.format = timeStringFormat;
    timeIntervalLabel.cellForRowTimeInteger = 0;
    timeIntervalLabel.drawRectTimeInteger = 0;
    [returnView bringSubviewToFront:timeIntervalLabel];
    returnView.dwuCellForRowTimeCountNumber = @(timeInterval
);
    return returnView;
}
}

static CollectionHeaderFooterBlock dwu_generateCollectionViewHe
aderFooterTimeLabel(SEL targetSelector, CGFloat labelWidth, NSStr
ing *timeStringFormat) {
    return ^(__unsafe_unretained id _self, __unsafe_unretained id
arg1, __unsafe_unretained id arg2, __unsafe_unretained id arg3
) {
        NSDate *date = [NSDate date];
        UIView *returnView = ((UIView * (^)(id, SEL, id, id))objc_
msgSend)(_self, targetSelector, arg1, arg2, arg3);
        NSTimeInterval timeInterval = ceilf([-date timeIntervals
inceNow] * 1000);
        [[returnView layer] dwu_scanLayerHierarchyRecursively];
        DWUKVOLabel *timeIntervalLabel = (DWUKVOLabel *)[returnV
iew viewWithTag:DWU_TIME_INTERVAL_LABEL_TAG];
        if (!timeIntervalLabel) {
            timeIntervalLabel = [[DWUKVOLabel alloc] initWithKVO
```

```
Target:returnView frame:CGRectMake(0, 0, labelWidth, DWU_LABEL_HEIGHT)];
    timeIntervalLabel.userInteractionEnabled = NO;
    timeIntervalLabel.backgroundColor = DWU_TEXT_LABEL_BACKGROUND_COLOR;
    timeIntervalLabel.textColor = DWU_TEXT_LABEL_FONT_COLOR;
    timeIntervalLabel.font = [UIFont boldSystemFontOfSize:DWU_LABEL_FONT_SIZE];
    timeIntervalLabel.textAlignment = NSTextAlignmentCenter;
    timeIntervalLabel.adjustsFontSizeToFitWidth = YES;
    timeIntervalLabel.tag = DWU_TIME_INTERVAL_LABEL_TAG;
    timeIntervalLabel.layer.dwuRecyclingCount++;
    [returnView addSubview:timeIntervalLabel];
}
timeIntervalLabel.format = timeStringFormat;
timeIntervalLabel.cellForRowTimeInteger = 0;
timeIntervalLabel.drawRectTimeInteger = 0;
[returnView bringSubviewToFront:timeIntervalLabel];
returnView.dwuCellForRowTimeCountNumber = @(timeInterval);
}
return returnView;
};
}

static void dwu_generateTimeLabelForUITableViewHeaderView()
{
    SEL selector = @selector(setDelegate:);
    NSString *selStr = NSStringFromSelector(selector);
    SEL newSelector = NSSelectorFromString([NSString stringWithFormat:@"dwu_uitableview_headerfooter_%@", selStr]);
    dwu_replaceMethodWithBlock(UITableView.class, selector, newSelector, ^(__unsafe_unretained UITableView *_self, __unsafe_unretained id arg) {
        SEL viewForHeaderInSectionSel = @selector(tableView:viewForHeaderInSection:);
        if ([arg respondsToSelector:viewForHeaderInSectionSel])
        {
            NSString *viewForSectionSelSelStr = NSStringFromSele
```

```
ctor(viewForHeaderInSectionSel);
    SEL newViewForSectionSel = NSSelectorFromString([NSString stringWithFormat:@"dwu_%@", viewForSectionSelSelStr]);
    dwu_replaceMethodWithBlock([arg class], viewForHeaderInSectionSel, newViewForSectionSel, dwu_generateTimeLabel(newViewForSectionSel, DWU_LABEL_WIDTH_UITABLEVIEW_CELL, DWU_LABEL_FORMAT_UITABLEVIEW_HEADER));
}
SEL viewForFooterInSectionSel = @selector(tableView:viewForFooterInSection:);
if ([arg respondsToSelector:viewForFooterInSectionSel])
{
    NSString *viewForSectionSelSelStr = NSStringFromSelector(viewForFooterInSectionSel);
    SEL newViewForSectionSel = NSSelectorFromString([NSString stringWithFormat:@"dwu_%@", viewForSectionSelSelStr]);
    dwu_replaceMethodWithBlock([arg class], viewForFooterInSectionSel, newViewForSectionSel, dwu_generateTimeLabel(newViewForSectionSel, DWU_LABEL_WIDTH_UITABLEVIEW_CELL, DWU_LABEL_FORMAT_UITABLEVIEW_FOOTER));
}
((void (*) (id, SEL, id))objc_msgSend)(_self, newSelector, arg);
});
}

static void dwu_generateTimeLabelForUITableviewCell() {
    SEL selector = @selector(setDataSource:);
    NSString *selStr = NSStringFromSelector(selector);
    SEL newSelector = NSSelectorFromString([NSString stringWithFormat:@"dwu_uitableview_%@", selStr]);
    dwu_replaceMethodWithBlock(UITableView.class, selector, newSelector, ^(__unsafe_unretained UITableView *_self, __unsafe_unretained id arg) {
        SEL cellForRowSel = @selector(tableView:cellForRowAtIndexPath:);
        NSString *cellForRowSelStr = NSStringFromSelector(cellForRowSel);
        SEL newCellForRowSel = NSSelectorFromString([NSString stringWithFormat:@"dwu_%@", cellForRowSelStr]);
    });
}
```

```
        dwu_replaceMethodWithBlock([arg class], cellForRowSel, newCellForRowSel, dwu_generateTimeLabel(newCellForRowSel, DWU_LABEL_WIDTH_UITABLEVIEW_CELL, DWU_LABEL_FORMAT_UITABLEVIEW_CELL));
        ((void ( *)(id, SEL, id))objc_msgSend)(_self, newSelector, arg);
    });
}

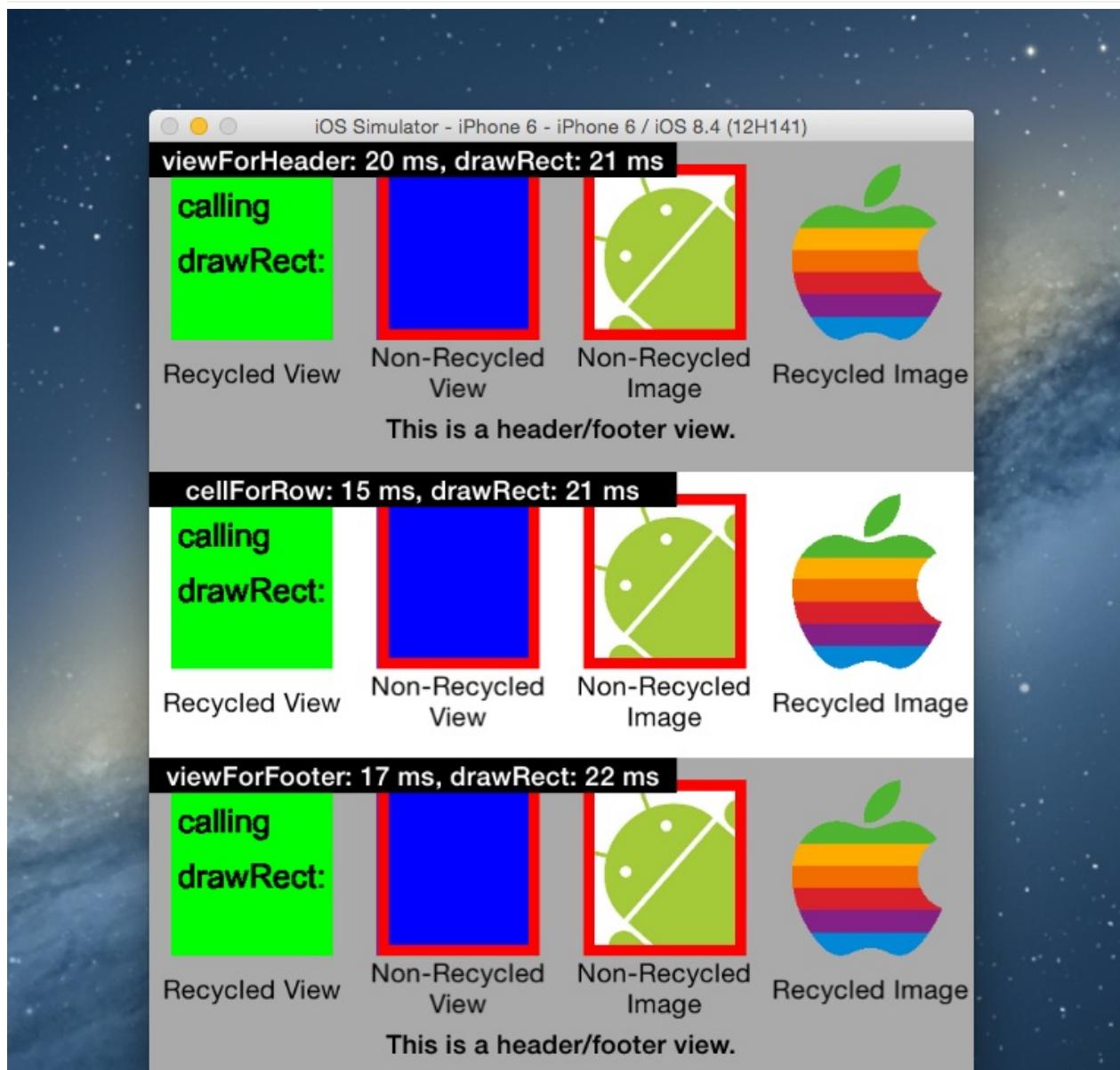
static void dwu_generateTimeLabelForUICollectionViewCell() {
    SEL selector = @selector(setDataSource:);
    NSString *selStr = NSStringFromSelector(selector);
    SEL newSelector = NSSelectorFromString([NSString stringWithFormat:@"dwu_uicollectionview_%@", selStr]);
    dwu_replaceMethodWithBlock(UICollectionView.class, selector, newSelector, ^(__unsafe_unretained UICollectionView *_self, __unsafe_unretained id arg) {
        SEL cellForItemSel = @selector(collectionView:cellForItemAtIndexPath:);
        NSString *cellForItemSelStr = NSStringFromSelector(cellForItemSel);
        SEL newCellForItemSel = NSSelectorFromString([NSString stringWithFormat:@"dwu_%@", cellForItemSelStr]);
        dwu_replaceMethodWithBlock([arg class], cellForItemSel, newCellForItemSel, dwu_generateTimeLabel(newCellForItemSel, DWU_LABEL_WIDTH_UICOLLECTIONVIEW_CELL, DWU_LABEL_FORMAT_UICOLLECTIONVIEW_CELL));

        cellForItemSel = @selector(collectionView:viewForSupplementaryElementOfKind:atIndexPath:);
        if ([arg respondsToSelector:@selector(cellForItemSel)]) {
            cellForItemSelStr = NSStringFromSelector(cellForItemSel);
            newCellForItemSel = NSSelectorFromString([NSString stringWithFormat:@"dwu_%@", cellForItemSelStr]);
            dwu_replaceMethodWithBlock([arg class], cellForItemSel, newCellForItemSel, dwu_generateCollectionViewHeaderFooterTimeLabel(newCellForItemSel, DWU_LABEL_WIDTH_UICOLLECTIONVIEW_CELL, DWU_LABEL_FORMAT_UICOLLECTIONVIEW_CELL));
        }
    });
}
```

```
        ((void *) (id, SEL, id))objc_msgSend) (_self, newSelector, arg);
    });
}

__attribute__((constructor)) static void DWUReturningAlert(void)
{
    @autoreleasepool {
        dwu_generateTimeLabelForUITableViewCell();
        dwu_generateTimeLabelForUITableViewHeaderFooterView();
        dwu_generateTimeLabelForUICollectionViewCell();
    }
}
#endif
```

## 效果图



## 备注

类似推荐

- GitHub - kconner/KMCGeigerCounter

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-21 | Alfred Jiang | -  |

## 方案名称

Xcode - 使用 Instruments 的 Allocations 工具检测内存分配情况

## 关键字

Xcode \ Instruments \ Allocations \ 内存分配

## 需求场景

1. 需要对 App 进行内存分配检测优化时
2. 《[Instruments 用户指南](#)》

## 参考链接

(见详细内容)

## 详细内容

以下链接可以帮助理解内存陷阱并给出了很好的解决方案参考

1. [使用 Swift 和 Objective-C 执行 iOS 内存管理的 7 个简单技巧](#)
2. [51CTO - iOS内存暴增问题追查与使用陷阱](#)
3. [Apple documentation - Finding Abandoned Memory](#)
4. [CocoaChina - iOS系类教程之用instruments来检验你的app:Instruments Tutorial with Swift: Getting Started 译文](#)
5. [iOS系类教程之用instruments来检验你的app](#)
6. [Ray Wenderlich - Instruments Tutorial with Swift: Getting Started](#)
7. [Segmentfault - iOS 性能优化：Instruments 工具的救命三招](#)

## 8. 解决MWPhotoBrowser中的SDWebImage加载大图导致的内存警告问题

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-26 | Alfred Jiang | -  |

## 方案名称

Xcode - 使用 Instruments 的 Core Animation 工具进行 UIKit 性能调优

## 关键字

Xcode \ Instruments \ Core Animation \ UIKit \ 性能调优

## 需求场景

1. 需要对 App 页面显示性能进行优化提高时

## 参考链接

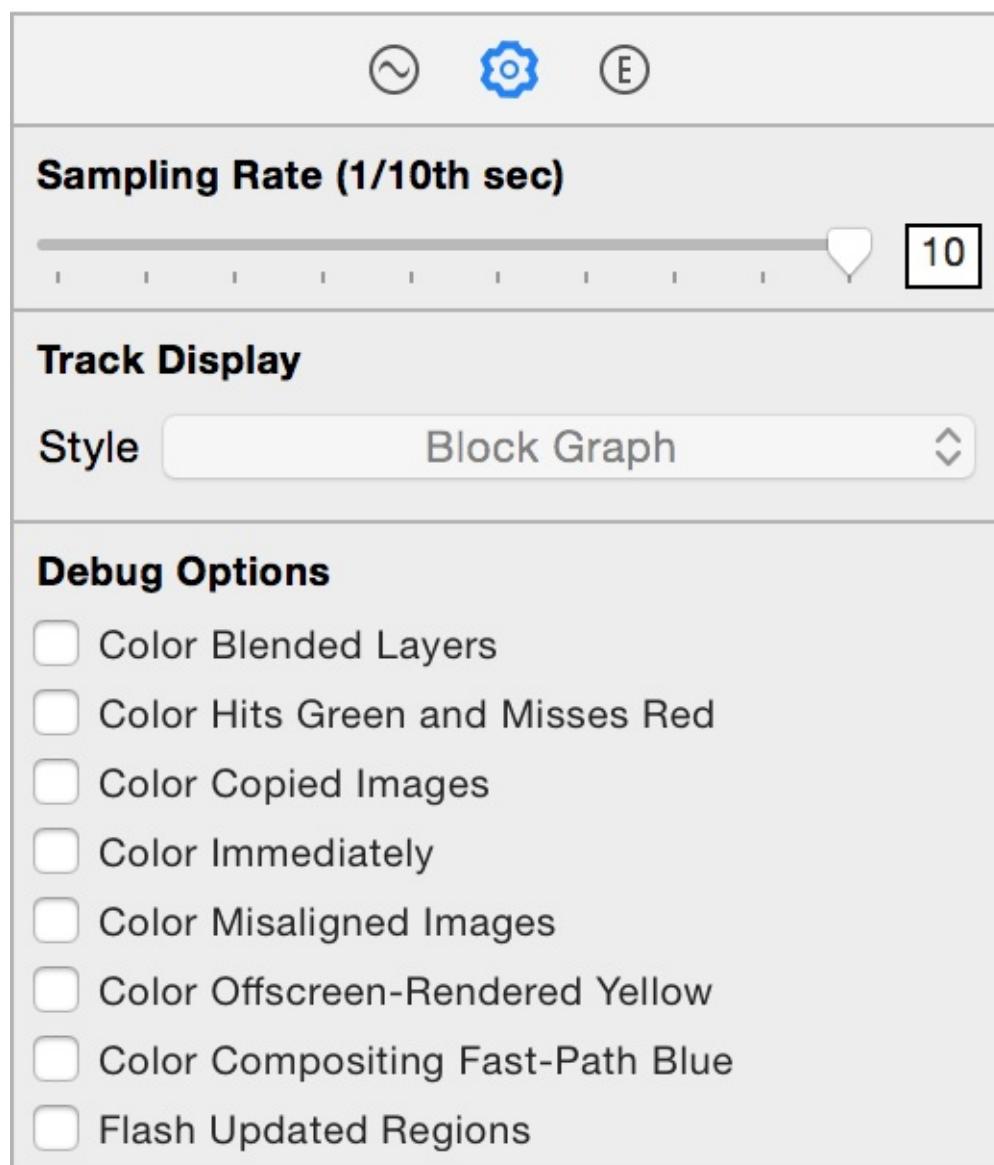
1. 简书 - [UIKit性能调优实战讲解](#)
2. [iOS App性能优化](#)
3. [《Instruments 用户指南》](#)

## 详细内容

### 1. 使用方法

在真机环境下打开 Instrument, 选择 Core Animation 工具进行调试

### 2. 调试项说明



| 调试项                             | 功能     | 说明                                   | 备注                                  |
|---------------------------------|--------|--------------------------------------|-------------------------------------|
| Color Blended Layers            | 检测图层混合 | 图层混合层会显示为红色，调优的方向是减少红色显示区域           | -                                   |
| Color Hits Green and Misses Red | 检测光栅化  | 如果命中缓存则显示为绿色，否则显示为红色，显然绿色越多越好，红色越少越好 | 光栅化是将一个layer预先渲染成位图(bitmap)，然后加入缓存中 |
|                                 | 检测图    | 如果存在需要转换格式的图                         | 应用中若出现GPU不                          |

|                                  |            |                                                 |                               |
|----------------------------------|------------|-------------------------------------------------|-------------------------------|
| Copied Images                    | 片格式转化      | 片，则会将图片标记为蓝色，调优的方向是减少蓝色显示区域                     | 支持的图片格式，则需要CPU预先进行格式转化才能正常显示。 |
| Color Immediately                | 取消颜色刷新延时   | 表示Instruments在做color-flush操作时取消10毫秒的延时          | -                             |
| Color Misaligned Images          | 检测图片压缩和对齐  | 如果图片需要缩放则标记为黄色，如果没有像素对齐则标记为紫色，调优的方向是减少黄色和紫色显示区域 | -                             |
| Color Offscreen-Rendered Yellow  | 检测图片离屏渲染   | 如果图片出现离屏渲染则标记为黄色，调优的方向是减少黄色显示区域                 | 我们需要尽可能避免离屏渲染                 |
| Color Compositing Fast-Path Blue | 标记由硬件绘制的路径 | 用于标记图片由硬件绘制的路径，用蓝色表示，蓝色越多越好。                    | -                             |
| Flash Updated                    | 标记发生重绘     | 重绘的区域会被标记为黄色，应该把需要重绘的区域                         |                               |

|                 |       |                      |   |
|-----------------|-------|----------------------|---|
| Updated Regions | 重绘的区域 | 尽可能缩小，调优的方向是减少黄色显示区域 | - |
|-----------------|-------|----------------------|---|

3. 更多详细内容见参考链接

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-02-29 | Alfred Jiang | -  |

## 方案名称

Xcode - 使用 Jenkins 进行 App 自动化打包

## 关键字

Xcode \ Jenkins \ 持续集成 \ 自动化打包

## 需求场景

1. 希望构建自动化打包的持续集成平台

## 参考链接

1. [Jenkins](#)
2. [Jenkins+GitHub+Xcode+fir搭了一个持续集成环境](#)
3. [Code Review - iOS 下如何自动化打包 App](#)
4. [阿里云构建Jenkins服务](#)
5. [Installing Jenkins on Ubuntu](#)
6. [CSDN - Jenkins用户配置](#)
7. [使用 Xcodebuild + Jenkins + Apache 做 iOS 持续集成](#)
8. [Segmentfault - Jenkins+Github持续集成](#)

## 详细内容

### 安装环境

服务器：阿里云 软件系统：Ubuntu 14.04 64位 硬件系统：CPU - 1核；内存 - 512 MB；带宽 - 1Mbps

## 1. 安装 JAVA 环境

创建 `/usr/java/` 目录并进入

```
$ sudo mkdir /usr/java/ $ cd /usr/java/
```

下载 `jdk-7u79-linux-x64.tar.gz` 并解压

```
wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/7u79-b15/jdk-7u79-linux-x64.tar.gz"  
sudo tar xzf jdk-7u79-linux-x64.tar.gz
```

设置 **JDK**，以后切换 **java** 可以使用 `update-alternatives --config java`

```
sudo update-alternatives --install /usr/bin/java java  
/usr/java/jdk1.7.0_79/bin/java 1 sudo update-alternatives --install /usr/bin/jar  
jar /usr/java/jdk1.7.0_79/bin/jar 1 sudo update-alternatives --install  
/usr/bin/javac javac /usr/java/jdk1.7.0_79/bin/javac 1 sudo update-alternatives  
--set jar /usr/java/jdk1.7.0_79/bin/jar sudo update-alternatives --set javac  
/usr/java/jdk1.7.0_79/bin/javac
```

设置 **bashrc** 或者 **bash\_profile**

```
sudo export JAVA_HOME=/usr/java/jdk1.7.0_79 sudo export  
JRE_HOME=${JAVA_HOME}/jre sudo export  
CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib sudo export  
PATH=${JAVA_HOME}/bin:${JRE_HOME}/bin:$PATH
```

检查安装

```
$ java -version
```

输出 `java version "1.7.0_79"` 说明 JAVA 环境安装配置成功

## 2. 安装 Jenkins

安装

```
wget -q -O - https://jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add  
- sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ >  
/etc/apt/sources.list.d/jenkins.list' sudo apt-get update sudo apt-get install  
jenkins
```

使用

下载安装完成默认启动 Jenkins 服务

如果 Jenkins 停止，尝试使用以下命令重启

```
$ sudo /etc/init.d/jenkins start
```

如果有更多安装疑问，参考 [Installing Jenkins on Ubuntu](#)

添加用户

依次进入 系统管理-> Configure Global Security -> 启用安全 -> 访问控制

安全域 - Jenkins 专有用户数据库

授权策略 - 安全矩阵

添加用户名，选择权限

保存并设置密码以及注册信息

添加完成

### 3. 插件管理

依次进入 系统管理-> 插件管理 -> 可更新 \ 可选插件 \ 已安装 \ 高级

若列表为空，进入 高级 -> 立即获取 可更新列表

若列表或插件更新安装失败，可以在 高级 中设置代理

### 4. 更多设置见参考链接

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 2  | 2015-12-24 | Alfred Jiang | -  |

## 方案名称

Xcode - 使用 LLDB 调试代码

## 关键字

Xcode \ 调试 \ Debug \ LLDB

## 需求场景

1. 利用 LLDB 调试代码

## 参考链接

1. 简书 - 小笨狼与LLDB的故事
2. Objc - Dancing in the Debugger — A Waltz with LLDB
3. The LLDB Debugger

## 详细内容

### **expression**

简写 `e`，别名有 `expr & p & print & call`

作用 1 是执行某个表达式。

```
// 改变颜色  
(lldb) expression -- self.view.backgroundColor = [UIColor redColor]  
// 刷新界面  
(lldb) expression -- (void)[CATransaction flush]
```

作用 2 是将返回值输出。

```
(lldb) expression -- self.view  
(UIView *) $1 = 0x00007fe322c18a10
```

## po

LLDB 为 `expression -O --` 别名，等价于 `expression -O --`，

作用是打印对象信息。

```
(lldb) po self.view  
<UIView: 0x7fb2a40344a0; frame = (0 0; 375 667); autoresize = W+  
H; layer = <CALayer: 0x7fb2a4018c80>>
```

## thread backtrace

简写 `bt`

作用是打印堆栈信息。

```
(lldb) bt
```

## thread return

作用是取消方法继续执行或者直接返回某个特定值。

```
thread return [<expr>]
```

更多内容参考 [简书 - 小笨狼与LLDB的故事](#)

## 效果图

(无)

## 备注

- Auto Layout - 约束冲突断点调试方法
- Xcode - 调试相关

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注               |
|----|------------|--------------|------------------|
| 1  | 2016-04-26 | Alfred Jiang | -                |
| 1  | 2016-05-05 | Alfred Jiang | 添加多条宏定义并使用逻辑与或操作 |

## 方案名称

Xcode - 在 Xcode 里添加和使用 Debug 宏定义

## 关键字

Xcode \ Debug 模式 \ 宏定义

## 需求场景

1. 需要为工程设置仅在 Debug 模式下有效的代码时

## 参考链接

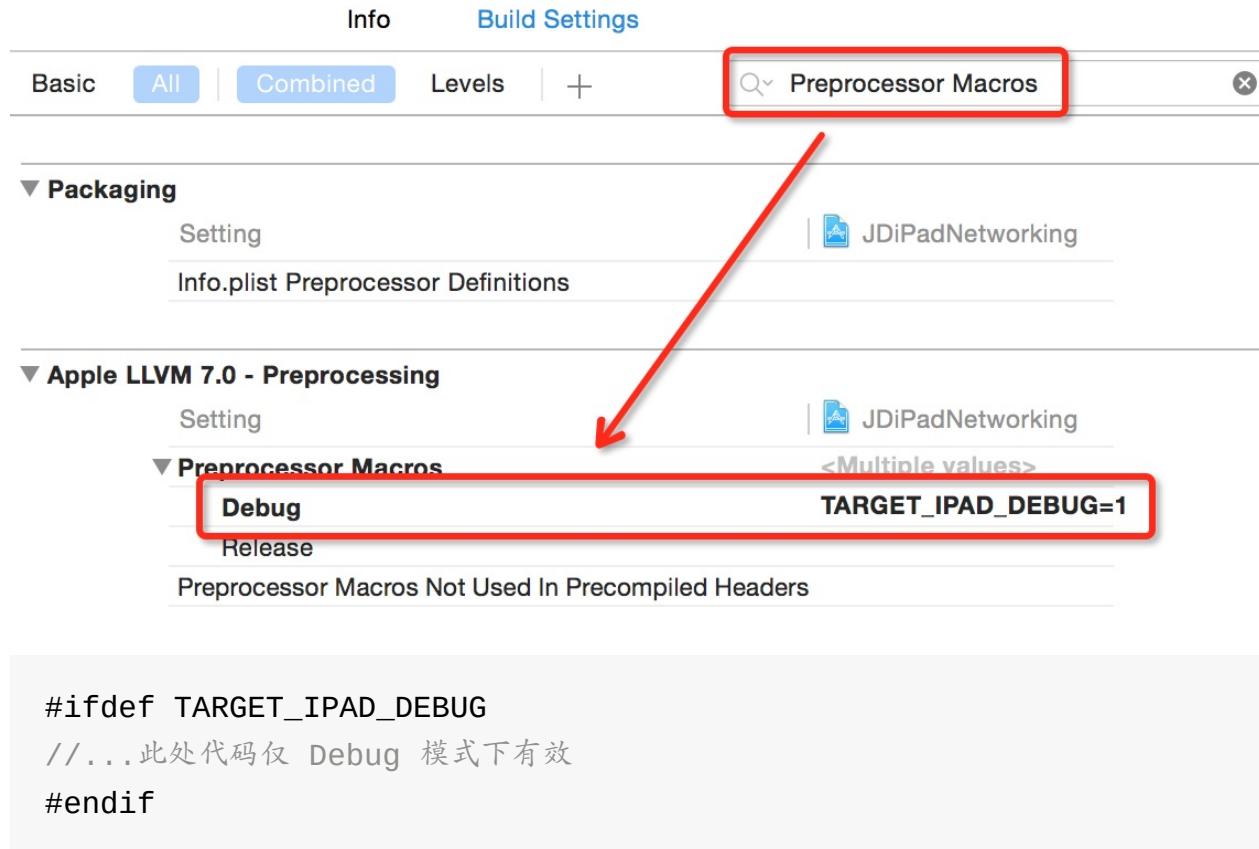
1. [Wutiam - #ifdef 中的逻辑与或操作](#)

## 详细内容

添加宏定义并使用

举例：添加 `TARGET_IPAD_DEBUG` 为 Debug 模式下特有宏定义

Xcode -> 工程 *Build Settings* -> All -> *Preprocessor Macros* ->  
`TARGET_IPAD_DEBUG=1`



添加多条宏定义并使用逻辑与或操作

```
#if (!defined A) && (defined B)
//... 此处代码仅 (!A && B) 时有效
#endif

#if (defined A) || (defined B)
//... 此处代码仅 (A || B) 时有效
#endif
```

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-29 | Alfred Jiang | -  |
| 2  | 2016-03-31 | Alfred Jiang | -  |

## 方案名称

Xcode - 插件开发学习备忘

## 关键字

Xcode \ 插件开发 \ 开发插件

## 需求场景

1. 开发自定义 Xcode 插件

## 参考链接

1. OneV's Den - Xcode 4 插件制作入门(推荐)
2. CocoaChina - Xcode 6 插件开发入门：添加自己的想法和功能
3. Forkong - Xcode7 插件开发：从开发到pull到Alcatraz
4. ManiacDev - Xcode Plugin Guide – Find Xcode Plugins
5. GitHub - Forkong/FKConsole
6. GitHub - macoscope/CodePilot(推荐)
7. GitHub - zulkis/ZKKeyBindingsTeacher

## 详细内容

### 1. 通过插件开发模板创建一个插件

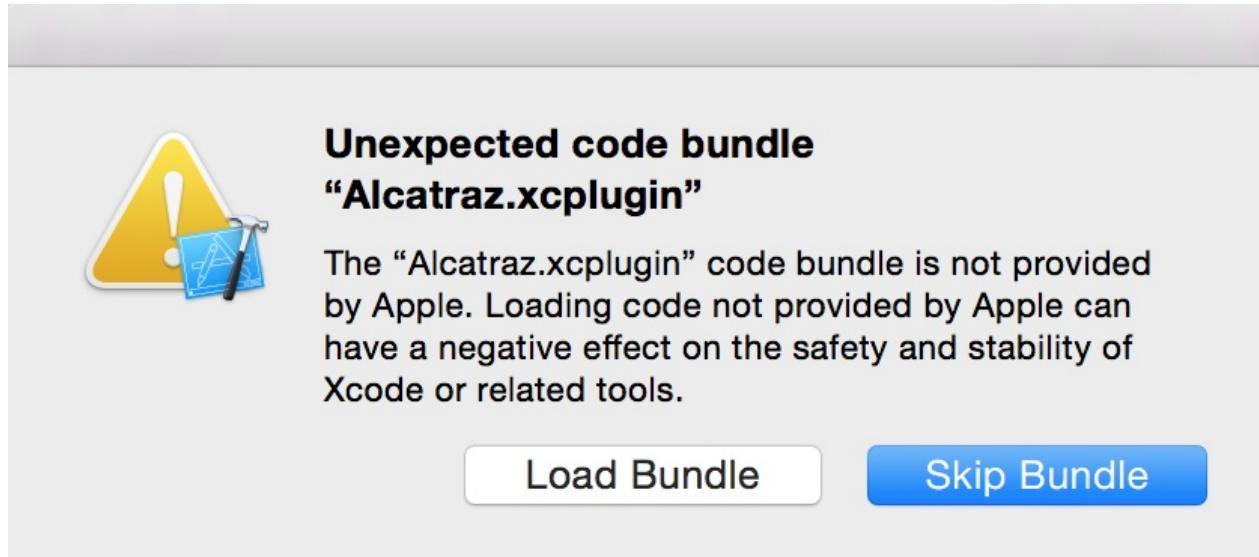
#### (1) 安装 Alcatraz

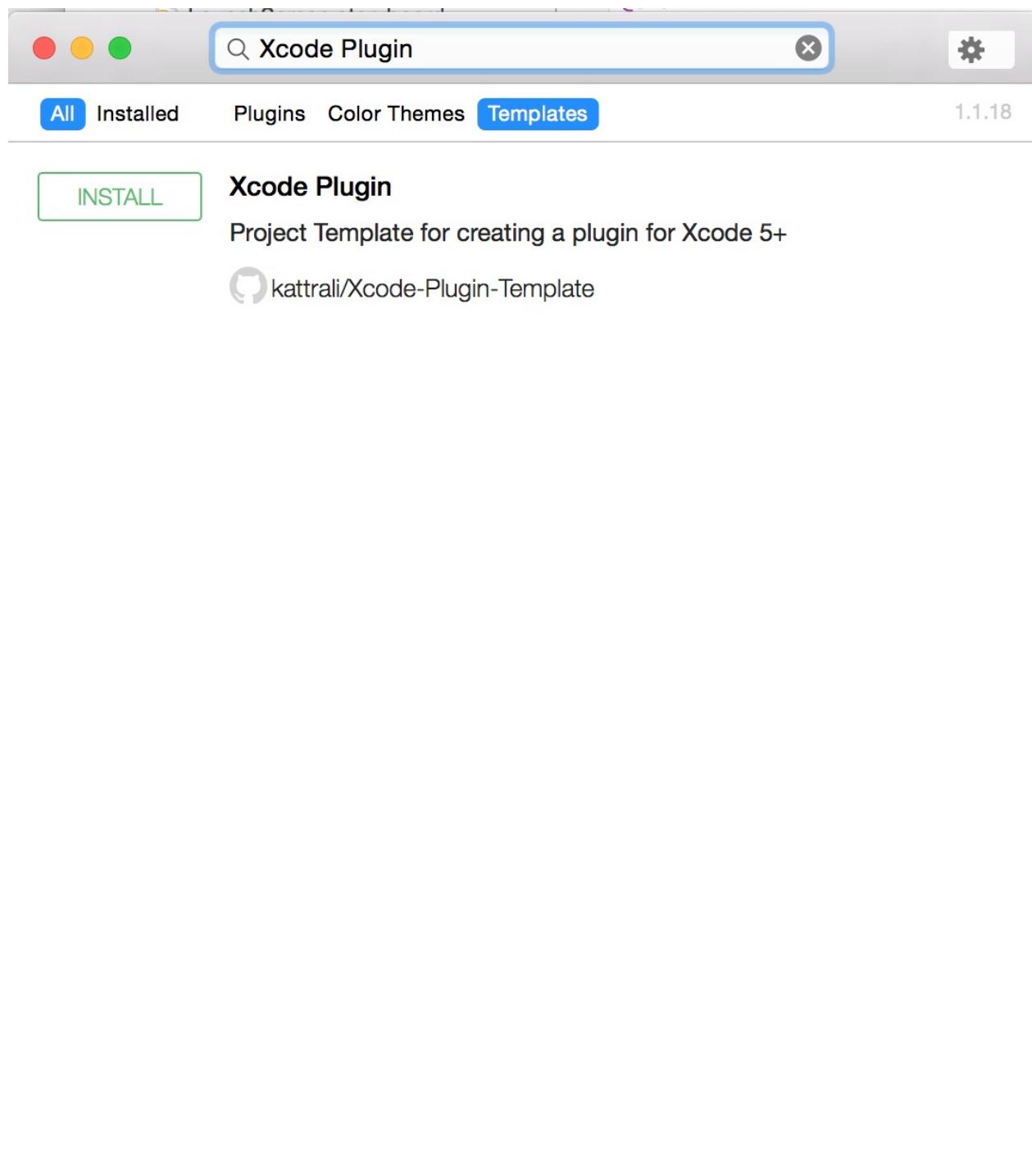
```
curl -fsSL
```

```
https://raw.githubusercontent.com/supermarin/Alcatraz/deploy/Scripts/install.sh | sh
```

(2) 重启 Xcode 选择 Load Bundle, 通过 Window -> Package Manager 安装

Xcode Plugin





注：

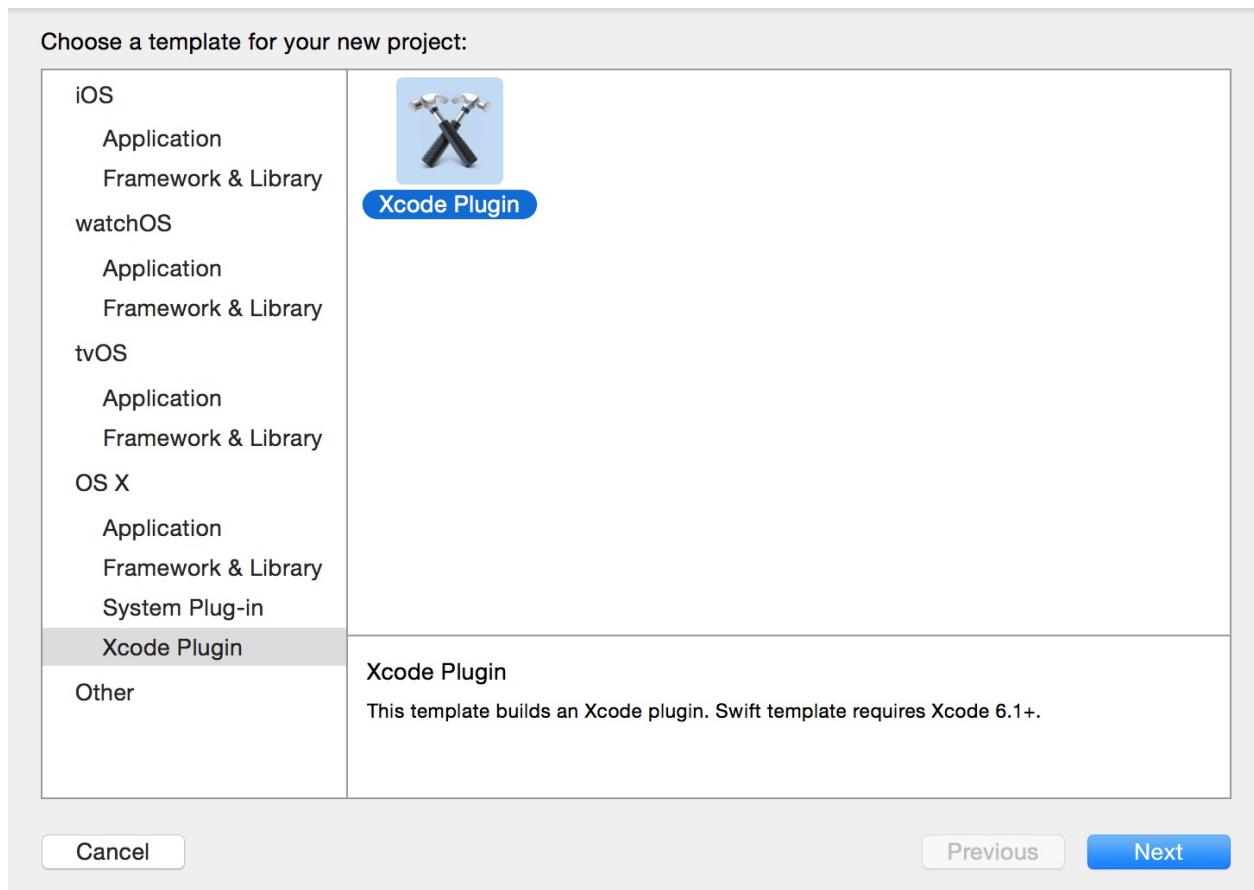
1. 如果无法通过 Alcatraz 安装 Xcode Plugin，可以在 GitHub 上直接下载 Xcode Plugin 编译运行，亦可自动安装插件开发模板
2. 安装 Xcode Plugin 前可通过以下命令获取 Xcode UUID，并在 info.plist -> DVTPPlugInCompatibilityUUIDs 中添加该 UUID, 否则无法正常安装

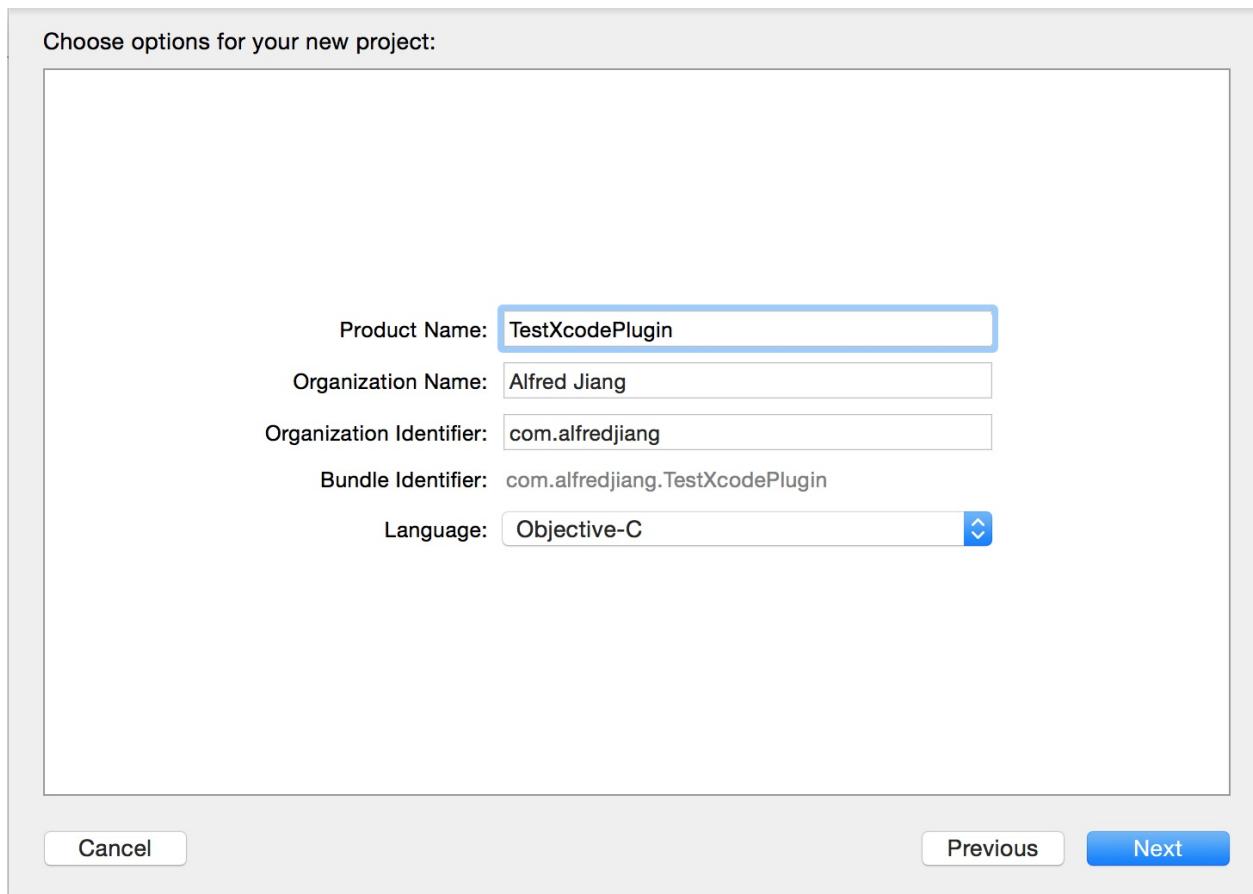
```
defaults read /Applications/Xcode.app/Contents/Info DVTPPlugInCompatibilityUUID
```

| Key                                  | Type       | Value                                 |
|--------------------------------------|------------|---------------------------------------|
| ▼ Information Property List          | Dictionary | (15 items)                            |
| Localization native development r... | String     | English                               |
| Executable file                      | String     | \$(EXECUTABLE_NAME)                   |
| Icon file                            | String     |                                       |
| Bundle identifier                    | String     | \$(PRODUCT_BUNDLE_IDENTIFIER)         |
| InfoDictionary version               | String     | 6.0                                   |
| Bundle name                          | String     | \$(PRODUCT_NAME)                      |
| Bundle OS Type code                  | String     | BNDL                                  |
| Bundle versions string, short        | String     | 1.0                                   |
| Bundle creator OS Type code          | String     | ????                                  |
| Bundle version                       | String     | 1                                     |
| ▼ DVTPluginCompatibilityUUIDs        | Array      | (14 items)                            |
| Item 0                               | String     | C4A681B0-4A26-480E-93EC-1218098B9AA0  |
| Item 1                               | String     | F41BD31E-2683-44B8-AE7F-5F09E919790E  |
| Item 2                               | String     | AD68E85B-441B-4301-B564-A45E4919A6AD  |
| Item 3                               | String     | A16FF353-8441-459E-A50C-B071F53F51B7  |
| Item 4                               | String     | 9F75337B-21B4-4ADC-B558-F9CADF7073A7  |
| Item 5                               | String     | E969541F-E6F9-4D25-8158-72DC3545A6C6  |
| Item 6                               | String     | 8DC44374-2B35-4C57-A6FE-2AD66A36AAD9  |
| Item 7                               | String     | AABB7188-E14E-4433-AD3B-5CD791EAD9A3  |
| Item 8                               | String     | 8DC44374-2B35-4C57-A6FE-2AD66A36AAD9  |
| Item 9                               | String     | AABB7188-E14E-4433-AD3B-5CD791EAD9A3  |
| Item 10                              | String     | 7FDF5C7A-131F-4ABB-9EDC-8C5F8F0B8A90  |
| Item 11                              | String     | 0420B86A-AA43-4792-9ED0-6FE0F2B16A13  |
| Item 12                              | String     | 7265231C-39B4-402C-89E1-16167C4CC990  |
| Item 13                              | String     | ACAA8656B-FEA8-4B6D-8E4A-93F4C95C362C |
| Minimum system version               | String     | \$(MACOSX_DEPLOYMENT_TARGET)          |
| Principal class                      | String     | TestXcodePlugin                       |
| XC4Compatible                        | Boolean    | YES                                   |
| XCPluginHasUI                        | Boolean    | NO                                    |

(3) 通过 File -> New -> Project -> Xcode Plugin 创建一个插件模板工程

### TestXcodePlugin





```

1 // 
2 // TestXcodePlugin.m
3 // TestXcodePlugin
4 //
5 // Created by viktyz on 16/4/3.
6 // Copyright © 2015年 Alfred Jiang. All rights reserved.
7 //
8 // Import header
9 #import "TestXcodePlugin.h"
10 @interface TestXcodePlugin()
11 @property (nonatomic, strong, readonly) NSBundle *bundle;
12 @end
13 @implementation TestXcodePlugin
14 + (instancetype)sharedPlugin
15 {
16     return sharedPlugin;
17 }
18 - (id)initWithBundle:(NSBundle *)plugin
19 {
20     if (self = [super init]) {
21         // Reference to plugin's bundle, for resource access
22         self.bundle = plugin;
23         [[NSNotificationCenter defaultCenter] addObserver:self
24                                         selector:@selector(didApplicationFinishLaunchingNotification:)
25                                         name:NSApplicationDidFinishLaunchingNotification
26                                         object:nil];
27     }
28     return self;
29 }
30 - (void)didApplicationFinishLaunchingNotification:(NSNotification *)notification
31 {
32     // Remove observer
33     [[NSNotificationCenter defaultCenter] removeObserver:self name:NSApplicationDidFinishLaunchingNotification object:nil];
34     // Create menu items, initialize UI, etc.
35     // Sample Menu Item:
36     NSMenuItem *menuItem = [[NSApp mainMenu] itemWithTitle:@"Edit"];
37     if (menuItem) {
38         [menuItem submenu addItem:[NSMenuItem separatorItem]];
39         NSMenuItem *actionMenuItem = [[NSMenuItem alloc] initWithTitle:@"Do Action" action:@selector(doMenuItem)
40                                     keyEquivalent:@""];
41         // Action MenuItem setKeyEquivalentModifierMask:NSAlphaShiftKeyMask | NSControlKeyMask;
42         [actionMenuItem setTarget:self];
43         [menuItem submenu addItem:actionMenuItem];
44     }
45 }
46 // Sample Action, for menu item:
47 - (void)doMenuItem
48 {
49     NSAlert *alert = [[NSAlert alloc] init];
50     [alert setMessageText:@"Hello, World!"];
51     [alert runModal];
52 }
53 - (void)dealloc
54 {
55     [[NSNotificationCenter defaultCenter] removeObserver:self];
56 }
57 @end

```

## 2. 为自定义插件添加快捷键支持

(1) 添加 `IDEKeyBindingPreferenceSet.h`, 该文件提供了 `Xcode` 快捷键绑定的私有接口

(2) 修改 TestXcodePlugin.m 文件如下

```
//  
// TestXcodePlugin.m  
// TestXcodePlugin  
//  
// Created by viktyz on 16/4/3.  
// Copyright © 2016年 Alfred Jiang. All rights reserved.  
//  
  
#import "TestXcodePlugin.h"  
#import "IDEKeyBindingPreferenceSet.h"  
  
  
  
#define CP_DEFAULT_SHORTCUT      @"$@X" // for key binding system  
  
#define DEFAULTS_KEY_BINDING     @"TestXcodePlugin.h"  
#define CP_MENU_PARENT_TITLE      @"Edit"  
#define CP_MENU_ITEM_TITLE        @"DoAction"  
  
  
  
static NSString * const IDEKeyBindingSetDidActivateNotification  
= @"IDEKeyBindingSetDidActivateNotification";  
  
  
  
@interface TestXcodePlugin()  
  
@property (nonatomic, strong, readwrite) NSBundle *bundle;  
@property (nonatomic, strong) NSMenuItem *actionMenuItem;  
  
@end  
  
@implementation TestXcodePlugin  
  
+ (instancetype)sharedPlugin  
{  
    return sharedPlugin;  
}
```

```

}

- (id)initWithBundle:(NSBundle *)plugin
{
    if (self = [super init]) {
        // reference to plugin's bundle, for resource access
        self.bundle = plugin;
        [[NSNotificationCenter defaultCenter] addObserver:self
                                                selector:@selector(didApplicationFinishLaunchingNotification)
                                              name:NSApplicationDidFinishLaunchingNotification
                                             object:nil];
    }
    return self;
}

- (void)didApplicationFinishLaunchingNotification:(NSNotification *)noti
{
    //removeObserver
    [[NSNotificationCenter defaultCenter] removeObserver:self name:NSApplicationDidFinishLaunchingNotification object:nil];

    [self setupKeyBindingsIfNeeded];
    [self installStandardKeyBinding];

    // Create menu items, initialize UI, etc.
    // Sample Menu Item:
    NSMenuItem *menuItem = [[NSApp mainMenu] itemWithTitle:CP_MENU_PARENT_TITLE];
    if (menuItem) {
        [[menuItem submenu] addItem:[NSMenuItem separatorItem]];
        self.actionMenuItem = [[NSMenuItem alloc] initWithTitle:CP_MENU_ITEM_TITLE action:@selector(doMenuAction) keyEquivalent:@""];
        //[[actionMenuItem setKeyEquivalentModifierMask:NSAlphaShiftKeyMask | NSControlKeyMask];
        [self.actionMenuItem setTarget:self];
        [[menuItem submenu] addItem:self.actionMenuItem];
    }
}

```

```

        [self updateMenuItem:self.actionMenuItem withShortcut:[s
elf keyboardShortcutFromUserDefaults]];
    }

    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(
keyBindingsHaveChanged:)
                                             name:IDEKeyBind
ingSetDidActivateNotification
                                             object:nil];
}

// Sample Action, for menu item:
- (void)doMenuAction
{
    NSAlert *alert = [[NSAlert alloc] init];
    [alert setMessageText:@"Hello, World"];
    [alert runModal];
}

- (void)dealloc
{
    [[NSNotificationCenter defaultCenter] removeObserver:self];
}

#pragma mark -

- (id<IDEKeyboardShortcut>)keyboardShortcutFromUserDefaults
{
    Class<IDEKeyboardShortcut> _IDEKeyboardShortcut = NSClassFro
mString(@"IDEKeyboardShortcut");
    return [_IDEKeyboardShortcut keyboardShortcutFromStringRepre
sentation:[self keyBindingFromUserDefaults]];
}

- (void)setupKeyBindingsIfNeeded
{
    if (IsEmpty([self keyBindingFromUserDefaults])) {
        [self saveKeyBindingToUserDefaults:CP_DEFAULT_SHORTCUT f
orKey:DEFAULTS_KEY_BINDING];
}

```

```
    }

}

- (NSString *)keyBindingFromUserDefaults
{
    return [[NSUserDefaults standardUserDefaults] valueForKey:DE
FAULTS_KEY_BINDING];
}

- (void)saveKeyBindingToUserDefaults:(NSString *)keyBinding forKey:(NSString *)defaultsKey
{
    [[NSUserDefaults standardUserDefaults] setObject:keyBinding
forKey:defaultsKey];
    [[NSUserDefaults standardUserDefaults] synchronize];
}

- (void)keyBindingsHaveChanged:(NSNotification *)notification
{
    [self updateKeyBinding:[self currentUserCPKeyBinding] forMen
uItem:self.actionMenuItem defaultsKey:DEFAULTS_KEY_BINDING];
}

- (void)updateKeyBinding:(id<IDEKeyBinding>)keyBinding forMenuItem:(NSMenuItem *)menuItem defaultsKey:(NSString *)defaultsKey
{
    if ([[keyBinding keyboardShortcuts] count] > 0) {
        id<IDEKeyboardShortcut> keyboardShortcut = [[keyBinding
keyboardShortcuts] objectAtIndex:0];
        [self saveKeyBindingToUserDefaults:[keyboardShortcut str
ingRepresentation] forKey:defaultsKey];
        [self updateMenuItem:menuItem withShortcut:keyboardShort
cut];
    }
}

- (void)updateMenuItem:(NSMenuItem *)menuItem withShortcut:(id<I
DEKeyboardShortcut>)keyboardShortcut
{
    [menuItem setKeyEquivalent:[keyboardShortcut keyEquivalent]]
}
```

```
;

    [menuItem setKeyEquivalentModifierMask:[keyboardShortcut modifierMask]];
}

- (id<IDEKeyBinding>)currentUserCPKeyBinding
{
    return [self menuKeyBindingWithItemTitle:CP_MENU_ITEM_TITLE
underMenuCalled:CP_MENU_ITEM_TITLE];
}

- (id<IDEMenuKeyBinding>)menuKeyBindingWithItemTitle:(NSString *)
itemTitle underMenuCalled:(NSString *)menuName
{
    Class<IDEKeyBindingPreferenceSet> _IDEKeyBindingPreferenceSet =
NSClassFromString(@"IDEKeyBindingPreferenceSet");

    id<IDEKeyBindingPreferenceSet> currentPreferenceSet = [[_IDE
KeyBindingPreferenceSet preferenceSetsManager] currentPreference
Set];

    id<IDEMenuKeyBindingSet> menuKeyBindingSet = [currentPrefere
nceSet menuKeyBindingSet] ;

    for (id<IDEMenuKeyBinding> keyBinding in [menuKeyBindingSet
keyBindings]) {
        if ([[keyBinding group] isEqualToString:menuName] && [[k
eyBinding title] isEqualToString:itemTitle]) {
            return keyBinding;
        }
    }

    return nil;
}

- (void)installStandardKeyBinding
{
    Class<IDEKeyBindingPreferenceSet> _IDEKeyBindingPreferenceSet =
NSClassFromString(@"IDEKeyBindingPreferenceSet");
```

```

    id<IDEKeyBindingPreferenceSet> currentPreferenceSet = [[_IDE
KeyBindingPreferenceSet preferenceSetsManager] currentPreference
Set];

    id<IDEMenuKeyBindingSet> menuKeyBindingSet = [currentPrefere
nceSet menuKeyBindingSet];

    Class<IDEKeyboardShortcut> _IDEKeyboardShortcut = NSClassFrom
String(@"IDEKeyboardShortcut");

    id<IDEKeyboardShortcut> defaultShortcut = [_IDEKeyboardShort
cut keyboardShortcutFromStringRepresentation:[self keyBindingFro
mUserDefaults]];

    Class<IDEMenuKeyBinding> _IDEMenuKeyBinding = NSClassFromStr
ing(@"IDEMenuKeyBinding");

    id<IDEMenuKeyBinding> cpKeyBinding = [_IDEMenuKeyBinding key
BindingWithTitle:CP_MENU_ITEM_TITLE

parentTitle:CP_MENU_PARENT_TITLE

group:CP_MENU_ITEM_TITLE

actions:[NSArray arrayWithObject:@"whatever:"]
k
eyboardShortcuts:[NSArray arrayWithObject:defaultShortcut]];

[cpKeyBinding setCommandIdentifier:CP_MENU_ITEM_TITLE];

[menuKeyBindingSet insertObject:cpKeyBinding inKeyBindingsAt
Index:0];
[menuKeyBindingSet updateDictionary];
}

#pragma mark -

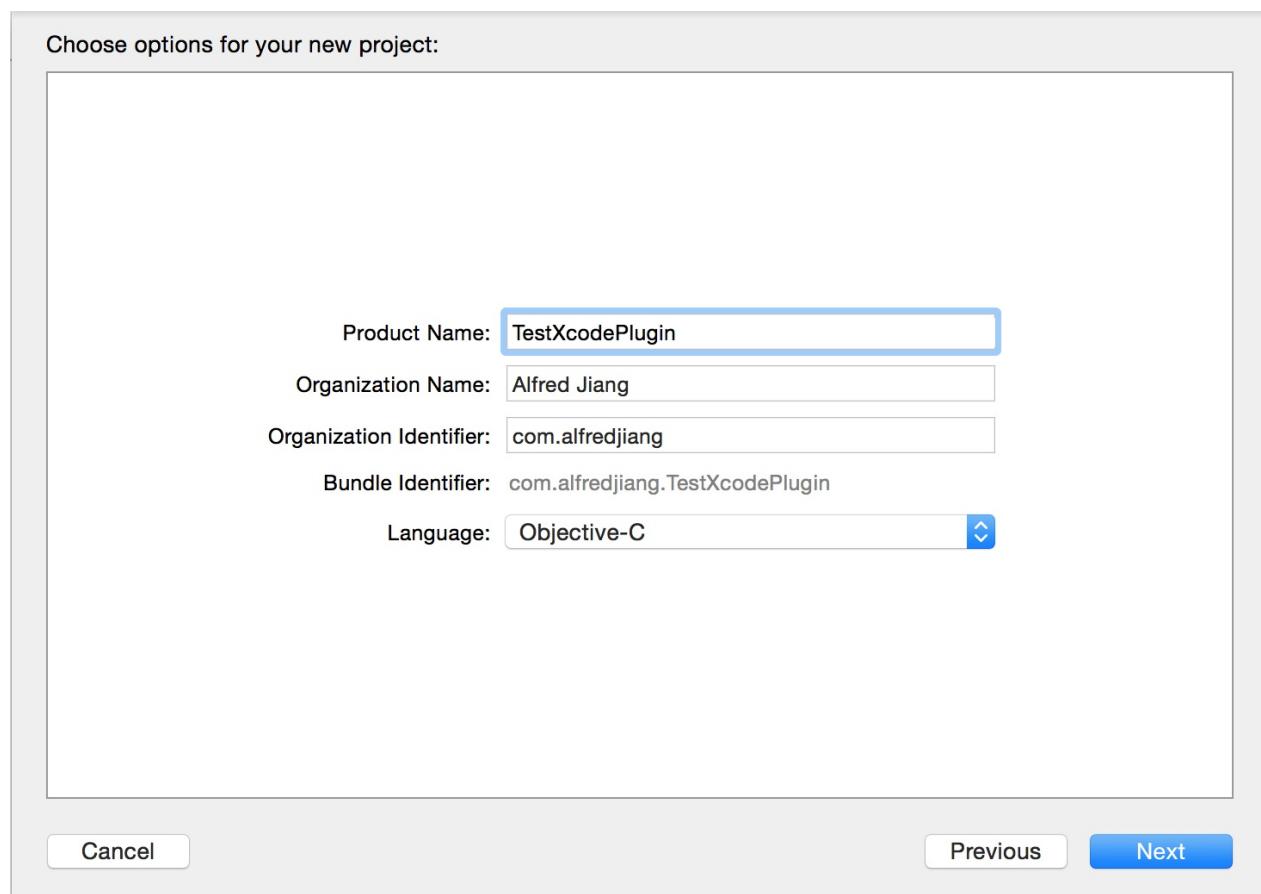
static inline BOOL IsEmpty(id thing) {
    return thing == nil
    || ([NSNull null]==thing)

```

```
    || ([thing respondsToSelector:@selector(length)] && [(NSData
*)thing length] == 0)
    || ([thing respondsToSelector:@selector(count)] && [(NSArray
*)thing count] == 0);
}

@end
```

(3) 修改后，原插件操作按钮旁边会显示快捷键，使用快捷键可触发插件相应事件



```

1 // TestXcodePlugin.m
2 // TestXcodePlugin
3 // Copyright © 2016年 Alfred Jiang. All rights reserved.
4 //
5 // Created by viktyz on 16/4/3.
6 // Copyright © 2016年 Alfred Jiang. All rights reserved.
7 //
8
9 #import "TestXcodePlugin.h"
10
11 @interface TestXcodePlugin()
12
13 @property (nonatomic, strong, readonly) NSBundle *bundle;
14 @end
15
16 @implementation TestXcodePlugin
17
18 + (instancetype)sharedPlugin
19 {
20     return sharedPlugin;
21 }
22
23 - (id)initWithBundle:(NSBundle *)plugin
24 {
25     if (self = [super init]) {
26         // reference to plugin's bundle, for resource access
27         self.bundle = plugin;
28         [[NSNotificationCenter defaultCenter] addObserver:self
29             selector:@selector(didApplicationFinishLaunchingNotification:)
30             name:NSApplicationDidFinishLaunchingNotification
31             object:nil];
32     }
33     return self;
34 }
35
36 - (void)didApplicationFinishLaunchingNotification:(NSNotification *)notification
37 {
38     //removeObserver
39     [[NSNotificationCenter defaultCenter] removeObserver:self name:NSApplicationDidFinishLaunchingNotification object:nil];
40
41     // Create menu items, initialize UI, etc.
42     NSMenuItem *menuItem = [[NSApp mainMenu] itemWithTitle:@"Edit"];
43     if (menuItem) {
44         [menuItem submenu] addItem:[NSMenuItem separatorItem];
45         NSMenuItem *actionMenuItem = [[NSMenuItem alloc] initWithTitle:@"Do Action" action:@selector(doMenuItemAction)
46             keyEquivalent:@""];
47         //actionMenuItem.setKeyEquivalentModifierMask:NSAlphaShiftKeyMask | NSControlKeyMask];
48         [actionMenuItem setTarget:self];
49         [actionMenuItem setIdentifier:1];
50         [menuItem submenu] addItem:actionMenuItem;
51     }
52
53     // Sample Action, for menu item:
54     -(void)doMenuItemAction
55     {
56         NSAlert *alert = [[NSAlert alloc] init];
57         [alert setMessageText:@"Hello, World"];
58         [alert runModal];
59     }
60
61     -(void)dealloc
62     {
63         [[NSNotificationCenter defaultCenter] removeObserver:self];
64     }
65
66 @end

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注                       |
|----|------------|--------------|--------------------------|
| 1  | 2015-03-25 | Alfred Jiang | -                        |
| 2  | 2015-12-10 | Alfred Jiang | 更新 Xcode 7 Alcatraz 安装相关 |
| 3  | 2015-12-23 | Alfred Jiang | -                        |

## 方案名称

Xcode - 插件的安装与使用

## 关键字

Xcode \ 插件 \ Alcatraz

## 需求场景

1. 提高 Xcode 开发效率

## 参考链接

1. 使用 Alcatraz 来管理 Xcode 插件
2. xcode 一些有用的小插件
3. 那些不能错过的 Xcode 插件
4. 简书 - 在 Xcode 7 中安装 Alcatraz
5. 升级 Xcode 之后 VVDocumenter-Xcode 不能用的解决办法
6. GitHub - alcatraz/Alcatraz

## 详细内容

### 1. Xcode 7 Alcatraz 管理包安装方法

(1) 关闭 Xcode。

(2) 如果你之前安装过 **Alcatraz**，卸载它。运行命令：

```
rm -rf ~/Library/Application\ Support/Developer/Shared/Xcode/Plug-ins/Alcatraz.xcplugin
```

(3) 运行命令：

```
find ~/Library/Application\ Support/Developer/Shared/Xcode/Plug-ins -name Info.plist -maxdepth 3 | xargs -l{} defaults write {} DVTPlugInCompatibilityUUIDs -array-add defaults read /Applications/Xcode.app/Contents/Info DVTPlugInCompatibilityUUID sudo xcode-select --reset
```

(4) 打开 **Xcode**。

(5) 安装 **Alcatraz**。

```
curl -fsSL https://raw.github.com/supermarin/Alcatraz/master/Scripts/install.sh | sh
```

(6) 重启 **Xcode**。若提示“**Load bundle**”、“**Skip Bundle**”，这里必须选择“**Load bundle**”，不然插件无法使用。

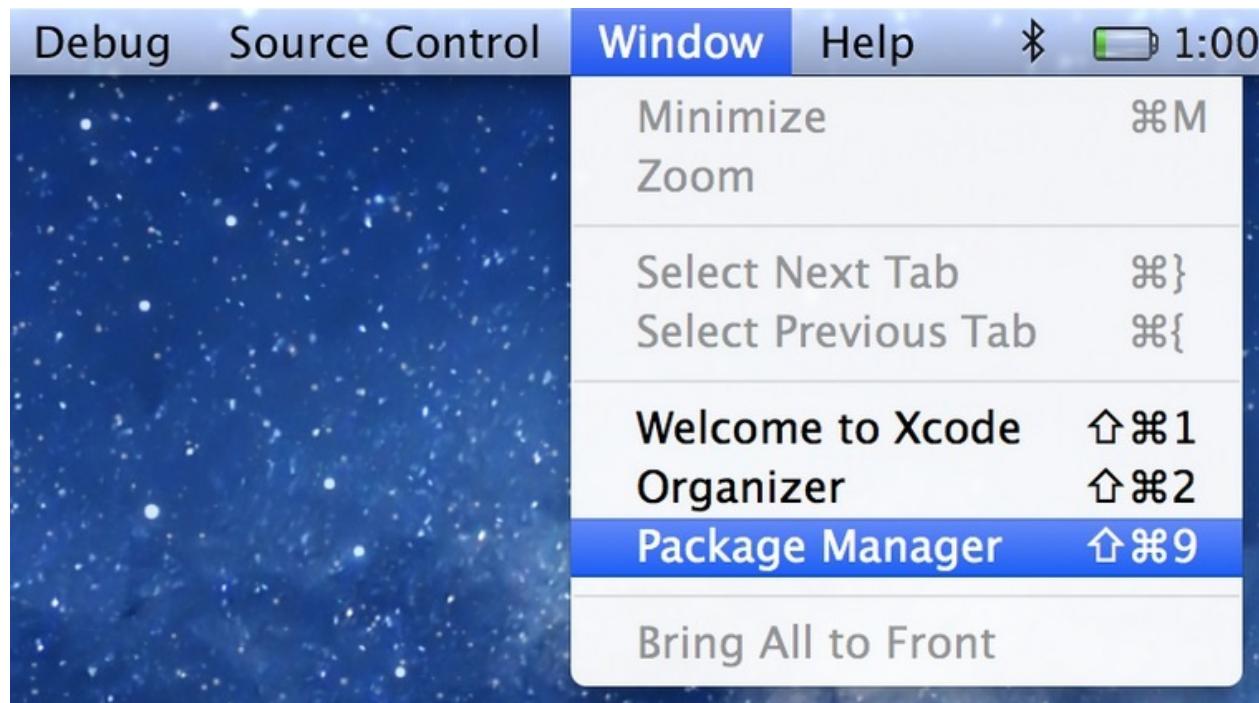
## 2. Xcode 7 之前使用如下的命令行来安装 **Alcatraz**

```
mkdir -p ~/Library/Application\ Support/Developer/Shared/Xcode/Plug-ins;
curl -L http://git.io/IOQWeA | tar xvz -C ~/Library/Application\ Support/Developer/Shared/Xcode/Plug-ins
```

卸载命令

```
rm -rf ~/Library/Application\ Support/Developer/Shared/Xcode/Plug-ins/Alcatraz.xcplugin rm -rf ~/Library/Application\ Support/Alcatraz
```

## 3. 在下面位置打开安装的 **Alcatraz** 插件管理器



#### 4. 选择喜欢的插件进行安装



##### **ClangFormat**



Xcode plug-in to use clang-format from in Xcode and consistently format your code with Clang

##### **Cobalt**



Cobalt color theme for Xcode

##### **CocoaPods**



CocoaPods integration right in Xcode

##### **CSSEdit Tribute**



CSSEdit Tribute color theme for Xcode

##### **DerivedData Exterminator**



Button for quickly deleting derived data. Makes Xcode happy.

## 5. 插件路径: *~/Library/Application Support/Developer/Shared/Xcode/Plug-ins/*

## 6. Xcode 6.2 插件失效解决方法

进入目录：*~/Library/Application Support/Developer/Shared/Xcode/Plug-ins/*

编辑各个插件目录下的Info.plist文件，找到DVTPlugInCompatibilityUUIDs这段，将新的UUID加进去，重启Xcode。

```
<string>A16FF353-8441-459E-A50C-B071F53F51B7</string>
```

## 7. 推荐插件

1. [FuzzyAutocompletePlugin](#) 代码自动补全
2. [VVDocumenter-Xcode](#) 快速生成注释
3. [KSImageNamed-Xcode](#) 在代码中预览图片
4. [SCXcodeSwitchExpander](#) 快速补全 Switch 的 Case

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-09-19 | Alfred Jiang | -  |

## 方案名称

Xcode - 系统警告处理( 清除和添加 )

## 关键字

Xcode \ 添加警告 \ 清除警告 \ Warning

## 需求场景

1. 消除系统中的编译警告

## 参考链接

1. [CocoaChina - 怎么去掉Xcode工程中的某种类型的警告](#)
2. [董铂然的博客 - iOS警告收录及科学快速的消除方法\(推荐\)](#)
3. [Which Clang Warning Is Generating This Message?](#)
4. [NSHipster - attribute](#)

## 详细内容

可以通过以下三种方式添加警告

1. 使用 `#warning TODO` 在代码中添加逻辑相关警告
2. 使用 `NS_DEPRECATED_IOS` 添加方法版本相关警告
3. 使用 `__attribute__((deprecated("警告描述")))` 添加方法详细描述警告

```
- (void)sampleWarningMethod __attribute__((deprecated("Unsafe method")));
```

常见警告以及对应清除方法可以参考[推荐链接](#)

如果某警告实在无法清除，但是又不想让它显示，可以加入预编译指令

```
#pragma clang diagnostic push  
#pragma clang diagnostic ignored "对应预编译指令"  
    //这里写出现警告的代码  
#pragma clang diagnostic pop
```

关于预编译指令的详细对应关系，可以[参考这里](#)

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注       |
|----|------------|--------------|----------|
| 1  | 2015-03-18 | Alfred Jiang | -        |
| 2  | 2015-12-07 | Alfred Jiang | 更新约束警告调试 |
| 3  | 2015-12-22 | Alfred Jiang | -        |
| 4  | 2016-01-04 | Alfred Jiang | -        |

## 方案名称

Xcode - 调试相关

## 关键字

Xcode \ Debug \ Crash

## 需求场景

1. 调试常见问题收集

## 参考链接

1. [Debugging iOS AutoLayout Issues](#)
2. [博客园 - iOS 各种调试技巧豪华套餐](#)

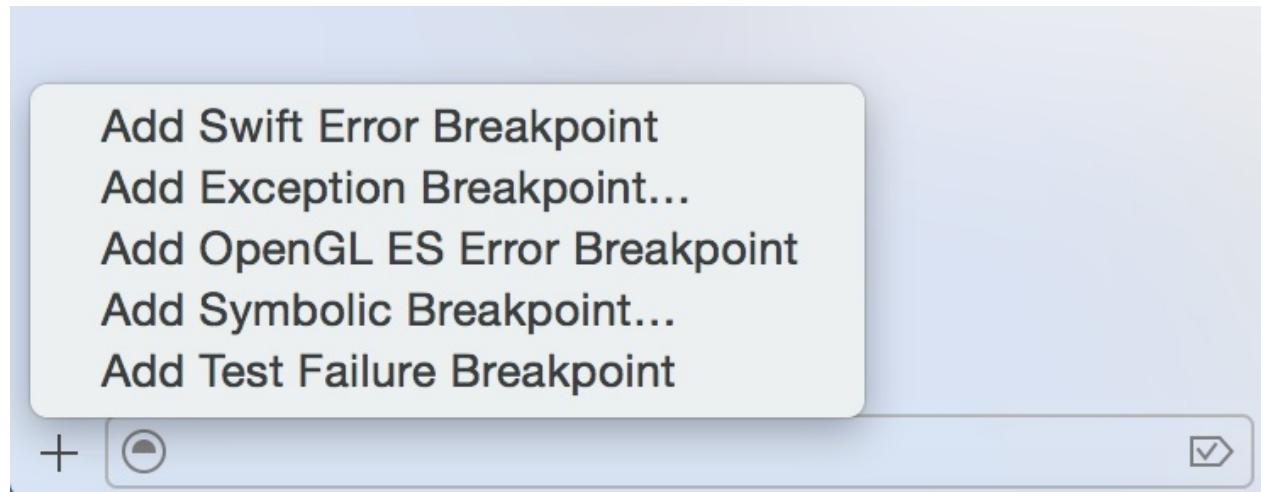
## 详细内容

### 1. unrecognized selector sent to instance 问题快速定位的方法

方法一：在 *Debug* 菜单中选择 *Breakpoints -> Create Symbolic Breakpoint*，在 *Symbol* 中填写如下方法签名：`-[NSObject(NSObject) doesNotRecognizeSelector:]`，然后再运行，错误时断点会停在真正导致崩溃的地方。

方法二：添加 *Exception* 类断点（推荐方案）

打开Xcode  
打开断点导航栏 cmd+7  
点击左下角 \*\*\*+\*\*\* 号按钮  
选择 Add Exception Breakpoint...

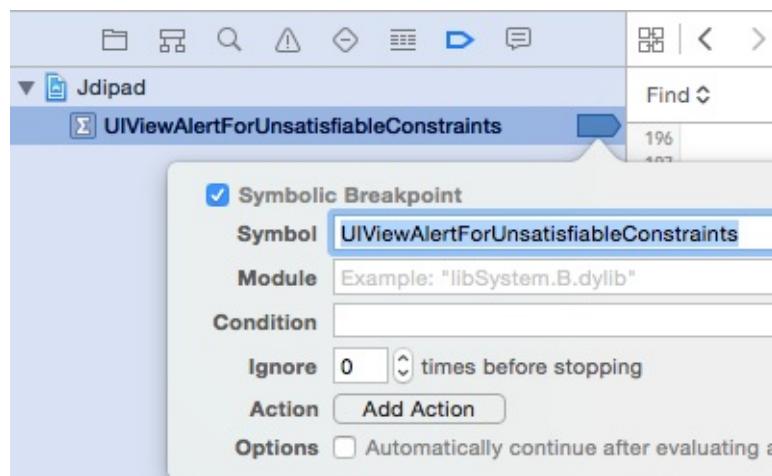


## 2. Swift Delegate Error show "use of undeclared type in swift project"

在 Swift 中，*Delegate* 名称不能和函数名相同

## 3. Make a symbolic breakpoint at UIAlertViewForUnsatisfiableConstraints to catch this in the debugger.

打开Xcode  
打开断点导航栏 cmd+7  
点击左下角 \*\*\*+\*\*\* 号按钮  
选择 Add Symbolic Breakpoint...  
右键 Edit Breakpoint...  
在 Symbol 中输入 UIAlertViewForUnsatisfiableConstraints



通过 打印内存地址信息可以查看控件信息

```
(lldb) po 0x7fc82aba1210
```

通过 recursiveDescription 方法可以查看全部页面层级关系

```
(lldb) po [[0x7fc82aba1210 superview] recursiveDescription]
```

#### 4. Break on void XXXxxx() to debug.

类似以下错误提示

```
This NSLayoutConstraint is being configured with a constant that exceeds internal limits. A smaller value will be substituted, but this problem should be fixed. Break on void _NSLayoutConstraintNumberExceedsLimit() to debug. This will be logged only once. This may break in the future.
```

添加断点方法

方法一：终端方式

点击暂停应用程序



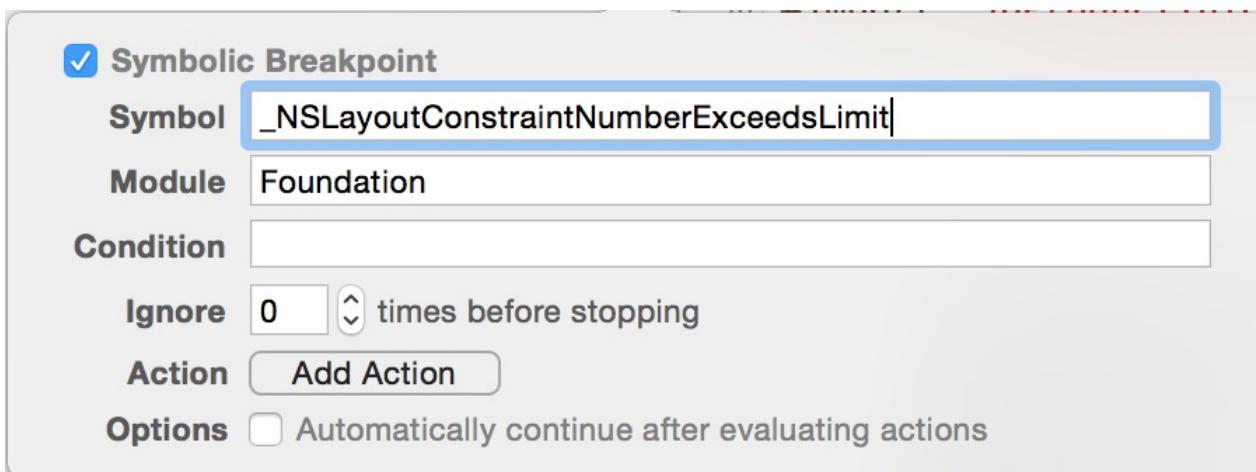
在终端输入

```
(lldb) b _NSLayoutConstraintNumberExceedsLimit  
Breakpoint 1: where = Foundation`_NSLayoutConstraintNumberExceedsLimit, address = 0x055e26cb
```

点击继续运行，再次进入会停在 crash 位置

方法二：添加 Xcode 断点

```
打开Xcode  
打开断点导航栏 cmd+7  
点击左下角 ***+*** 号按钮  
选择 Add Symbolic Breakpoint...  
右键 Edit Breakpoint...  
在 Symbol 中输入 _NSLayoutConstraintNumberExceedsLimit  
在 Module 中输入 Foundation
```



点击继续运行，再次进入会停在 crash 位置

## 5. `viewDidLayoutSubviews` 在 iOS 7 上导致应用崩溃

在 iOS 8 中使用 `viewDidLayoutSubviews`，应用正常运行，没有问题，但是应用在 iOS 7 上运行的时候，报错，导致应用崩溃，错误信息类似：

```
Cannot find executable for CFBundle 0x78f8f220 </Library/Developer/CoreSimulator/Profiles/Runtimes/iOS 7.1.simruntime/Contents/Resources/RuntimeRoot/System/Library/AccessibilityBundles/GeoServices.axbundle
```

```
Assertion failure in -[UIView layoutSublayersOfLayer:]....unrecognizeSelector.....
```

在 `viewDidLayoutSubviews` 方法末尾添加 `[self.view layoutSubviews]` 即可解决

## 效果图

(无)

## 备注

- Xcode - 使用 LLDB 调试代码
- Xcode - 运行时环境变量( Environment Variables )
- Auto Layout - 约束冲突断点调试方法

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-02-05 | Alfred Jiang | -  |

## 方案名称

Xcode - 运行时环境变量( Environment Variables )

## 关键字

Xcode \ 运行时 \ 环境变量 \ Environment Variables

## 需求场景

1. 添加环境变量可以帮助提高调试效率，如调试僵尸对象

## 参考链接

1. [运行时环境变量](#)
2. [Stack Overflow - List of environment variable in Objective C?](#)
3. [iOS环境变量设置](#)
4. [GitHub Gist - OBJC\\_HELP=1 on Lion](#)
5. [Apple documentation - Mac OS X Debugging Magic](#)

## 详细内容

添加方法：

Project -> Scheme -> Edit Scheme，在 Arguments 下可以添加运行时的环境变量( Environment Variables )

查看列表：打开终端执行以下命令可查看完整列表

```
$ export OBJC_HELP=1
$ /Applications/Safari.app/Contents/MacOS/Safari
```

列表说明：

| 变量名                           | 介绍                                                                     |            |
|-------------------------------|------------------------------------------------------------------------|------------|
| OBJC_PRINT_OPTIONS            | list which options are set                                             | 输出         |
| OBJC_PRINT_IMAGES             | log image and library names as they are loaded                         | 输出         |
| OBJC_PRINT_LOAD_METHODS       | log calls to class and category +load methods                          | 打的         |
| OBJC_PRINT_INITIALIZE_METHODS | log calls to class +initialize methods                                 | (v)        |
| OBJC_PRINT_RESOLVED_METHODS   | log methods created by +resolveClassMethod and +resolveInstanceMethod: | +re<br>+re |
| OBJC_PRINT_CLASS_SETUP        | log progress of class and category setup                               | 打          |
| OBJC_PRINT_PROTOCOL_SETUP     | log progress of protocol setup                                         | 打          |
| OBJC_PRINT_IVAR_SETUP         | log processing of non-fragile ivars                                    | 打          |
| OBJC_PRINT_VTABLE_SETUP       | log processing of class vtables                                        | 打          |
| OBJC_PRINT_VTABLE_IMAGES      | print vtable images showing overridden methods                         | 打          |
| OBJC_PRINT_CACHE_SETUP        | log processing of method caches                                        | 打          |
| OBJC_PRINT_FUTURE_CLASSES     | log use of future classes for toll-free bridging                       | 打到多        |
| OBJC_PRINT_GC                 | log some GC operations                                                 | 打          |

|                                 |                                                                                       |          |
|---------------------------------|---------------------------------------------------------------------------------------|----------|
| OBJC_PRINT_PREOPTIMIZATION      | log preoptimization courtesy of dyld shared cache                                     | 打        |
| OBJC_PRINT_CXX_CTORS            | log calls to C++ ctors and dtors for instance variables                               | 打        |
| OBJC_PRINT_EXCEPTIONS           | log exception handling                                                                |          |
| OBJC_PRINT_EXCEPTION_THROW      | log backtrace of every objc_exception_throw()                                         | 打        |
| OBJC_PRINT_ALT_HANDLERS         | log processing of exception alt handlers                                              | 打        |
| OBJC_PRINT_REPLACED_METHODS     | log methods replaced by category implementations                                      | 打        |
| OBJC_PRINT_DEPRECATED_WARNINGS  | warn about calls to deprecated runtime functions                                      | 打        |
| OBJC_PRINT_POOL_HIGHWATER       | log high-water marks for autorelease pools                                            | 打        |
| OBJC_PRINT_CUSTOM_RR            | log classes with unoptimized custom retain/release methods                            | 打<br>re  |
| OBJC_PRINT_CUSTOM_AWZ           | log classes with unoptimized custom allocWithZone methods                             | 打<br>all |
| OBJC_PRINT_RAW_ISA              | log classes that require raw pointer isa fields                                       | 打        |
| OBJC_DEBUG_UNLOAD               | warn about poorly-behaving bundles when unloaded                                      |          |
| OBJC_DEBUG_FRAGILE_SUPERCLASSES | warn about subclasses that may have been broken by subsequent changes to superclasses | 当        |
| OBJC_DEBUG_FINALIZERS           | warn about classes that implement -dealloc but not -finalize                          | 警        |
| OBJC_DEBUG_NIL_SYNC             | warn about @synchronized(nil),                                                        | 警        |

|                              |                                                                   |    |
|------------------------------|-------------------------------------------------------------------|----|
| OBJC_DEBUG_NIL_SYNC          | which does no synchronization                                     | 调  |
| OBJC_DEBUG_NONFRAGILE_IVARS  | capriciously rearrange non-fragile ivars                          | 不  |
| OBJC_DEBUG_ALT_HANDLERS      | record more info about bad alt handler use                        | 记  |
| OBJC_DEBUG_MISSING_POOLS     | warn about autorelease with no pool in place, which may be a leak | 警使 |
| OBJC_DEBUG_DUPLICATE_CLASSES | halt when multiple classes with the same name are present         |    |
| OBJC_USE_INTERNAL_ZONE       | allocate runtime data in a dedicated malloc zone                  | 在  |
| OBJC_DISABLE_GC              | force GC OFF, even if the executable wants it on                  | 强即 |
| OBJC_DISABLE_VTABLES         | disable vtable dispatch                                           |    |
| OBJC_DISABLE_PREOPTIMIZATION | disable preoptimization courtesy of dyld shared cache             | 关  |
| OBJC_DISABLE_TAGGED_POINTERS | disable tagged pointer optimization of NSNumber et al.            |    |
| OBJC_DISABLE_NONPOINTER_ISA  | disable non-pointer isa fields                                    | 关  |

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-09-28 | Alfred Jiang | -  |

## 方案名称

Xcode - 非 App Store 下载官方 Xcode 安装包方法

## 关键字

Xcode \ dmg \ Xcode 安装包

## 需求场景

1. 需要同时安装多个 Xcode 时
2. 需要安装较老版本 Xcode 时

## 参考链接

1. [CSDN - XCode 7.3.1\(dmg\) 官方直接下载地址\(离线下载\)](#) (推荐)
2. [Apple - Downloads for Apple Developers](#)

## 详细内容

当需要安装多个版本 Xcode 用于适配不同模拟器时，App Store 的唯一最新版往往难以满足需求，这时候需要进入[苹果的开发者中心](#)下载独立的安装包

### Xcode 8

- [8.0 下载地址](#)

### Xcode 7

- [7.3.1 下载地址](#)
- [7.0 下载地址](#)

## Xcode 6

- [6.4 下载地址](#)
- [6.0.1 下载地址](#)

更多苹果官方开发工具下载参考 [Apple - Downloads for Apple Developers](#)

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-08-24 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |
| 2  | 2016-03-11 | Alfred Jiang | -  |

## 方案名称

Xib - 通过 Xib 加载自定义 Cell

## 关键字

Xib \ UITableViewCell \ 自定义

## 需求场景

1. 需要实现较为复杂的 cell 布局，通过 xib 方式实现

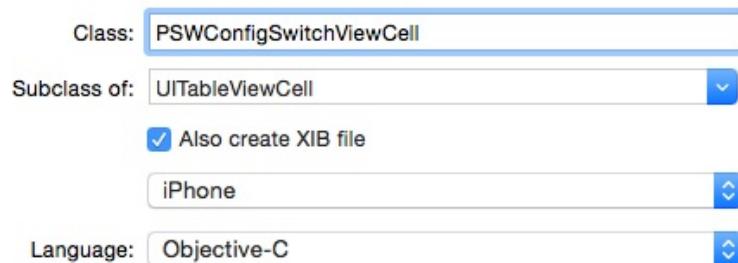
## 参考链接

1. 简书 - iOS 自定义Cell-通过XIB方式

## 详细内容

**2016-03-11 更新**

1. 通过 File -> New -> File... -> iOS -> Source -> Cocoa Touch Class 创建 Cell
2. 在 Class 中输入自定义 Cell 名称(示例：PSWConfigSwitchViewCell)；在 Subclass of 中输入父类 UITableViewCell；勾选 Also create Xib file 选项；选择平台 (iPhone / iPad) 和语言 (Objective-C / Swift)



### 3. 在 PSWConfigSwitchViewCell.h 中申明初始化方法

```
+ (instancetype)cellWithTableView:(UITableView *)tableView;
```

### 1. 在 PSWConfigSwitchViewCell.m 中实现初始化方法

```
+ (instancetype)cellWithTableView:(UITableView *)tableView
{
    static NSString *identifier = @"PSWConfigSwitchViewCell";

    PSWConfigSwitchViewCell *cell = [tableView dequeueReusableCellWithIdentifier:identifier];

    if (cell == nil) {
        cell = [[[NSBundle mainBundle] loadNibNamed:@"PSWConfigSwitchViewCell" owner:nil options:nil] firstObject];
    }

    return cell;
}
```

### 1. 在 cellForRowAtIndexPath 方法中加载自定义 Cell

```

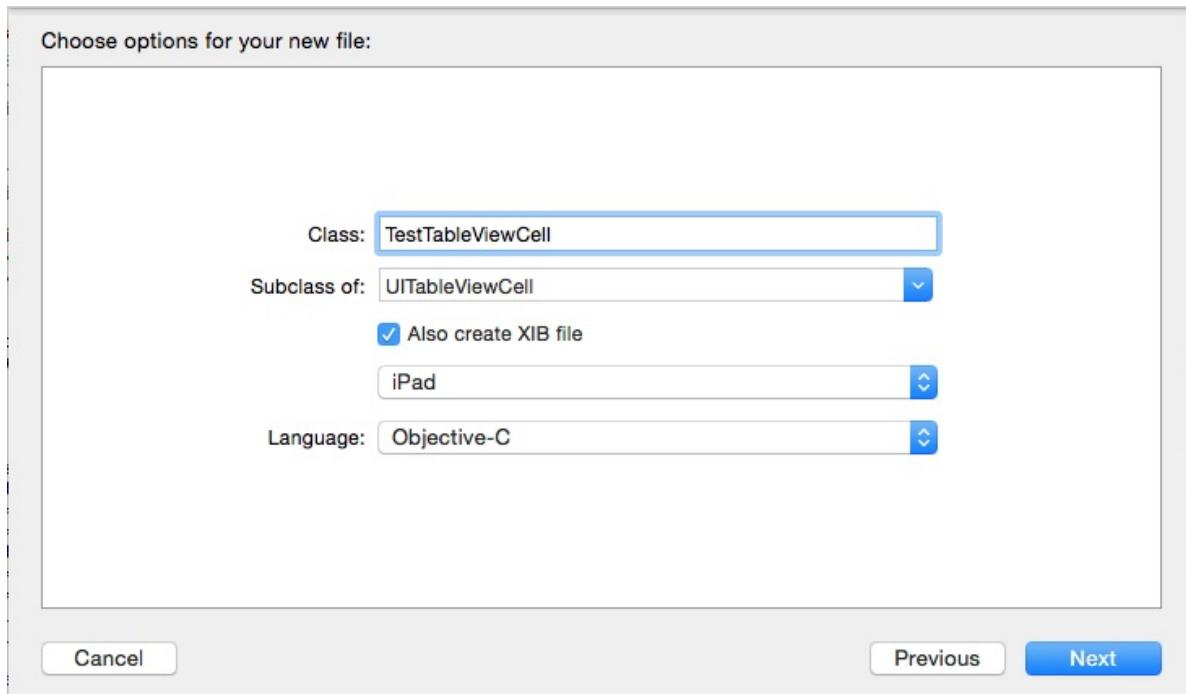
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    PSWConfigSwitchViewCell *cell = [PSWConfigSwitchViewCell cellWithTableView:tableView];
    // Config Your Cell
    return cell;
}

```

- 可以在 PSWConfigSwitchViewCell.xib 中布局更多控件并关联 PSWConfigSwitchViewCell 类，亦可以在 PSWConfigSwitchViewCell.(h/m) 中申明实现更多自定义方法。

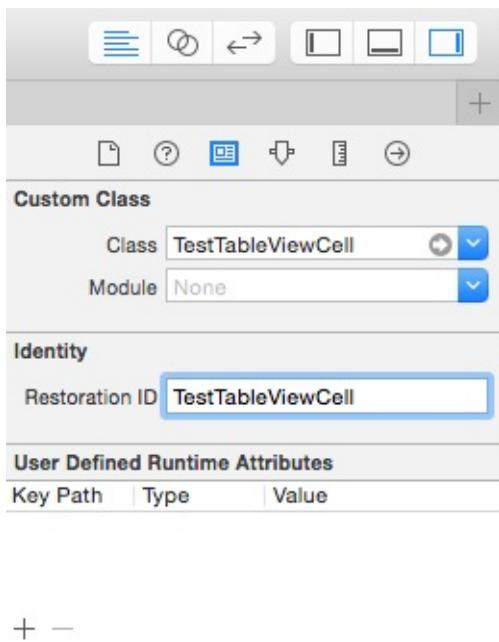
**2015-12-23** 之前创建

- 通过 File -> New -> File... -> iOS -> Source -> Cocoa Touch Class 创建 Cell
- 在 Class 中输入自定义 Cell 名称(示例：TestTableViewCell)；在 Subclass of 中输入父类 UITableViewCell；勾选 Also create Xib file 选项；选择平台 (iPhone / iPad) 和语言 (Objective-C / Swift)



- 下面进入 TestTableViewCell.xib -> Test TableView Cell -> Utilities 面板 -> Show the Identity inspector

4. 在 Restoration ID 中输入重用 ID(示例 : TestTableViewCell), 这个 ID 将用于重用 cell



5. 在 ViewDidLoad 方法中向使用该 Cell 的 UITableView 注册 Xib 定义的 Cell.

```
[self.tableViewMain registerNib:[UINib nibWithNibName:@"TestTableViewCell" bundle:nil] forCellReuseIdentifier:@"TestTableViewCell"];
```

1. 在 cellForRowAtIndexPath 方法中加载自定义 Cell

```

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *aTestTableViewCell = @"TestTableViewCell";
    TestTableViewCell *cell = (TestTableViewCell*)[tableView dequeueReusableCellWithIdentifier:aTestTableViewCell];
    if (!cell)
    {
        NSArray *nib = [[NSBundle mainBundle] loadNibNamed:@"TestTableViewCell" owner:[TestTableViewCell class] options:nil];
        cell = (TestTableViewCell *)[nib objectAtIndex:0];
        //这里的 0 表示 xib 中第 1 个View，也就是 Test TableView Cell
    }
    cell.selectionStyle = UITableViewCellSelectionStyleNone;
    ...
    return cell;
}

```

## 效果图

(无)

## 备注

尽量避免在 `cell` 的数据加载中进行数据运算，相关 `cell` 的显示依赖数据，尽量在 `Model` 中初始化完成，`cell` 直接读取显示。更多参考

- [UITableView - 滑动加载性能优化总结](#)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-08-24 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

Xib - 通过 Xib 加载自定义 View

## 关键字

Xib \ UIView \ 自定义

## 需求场景

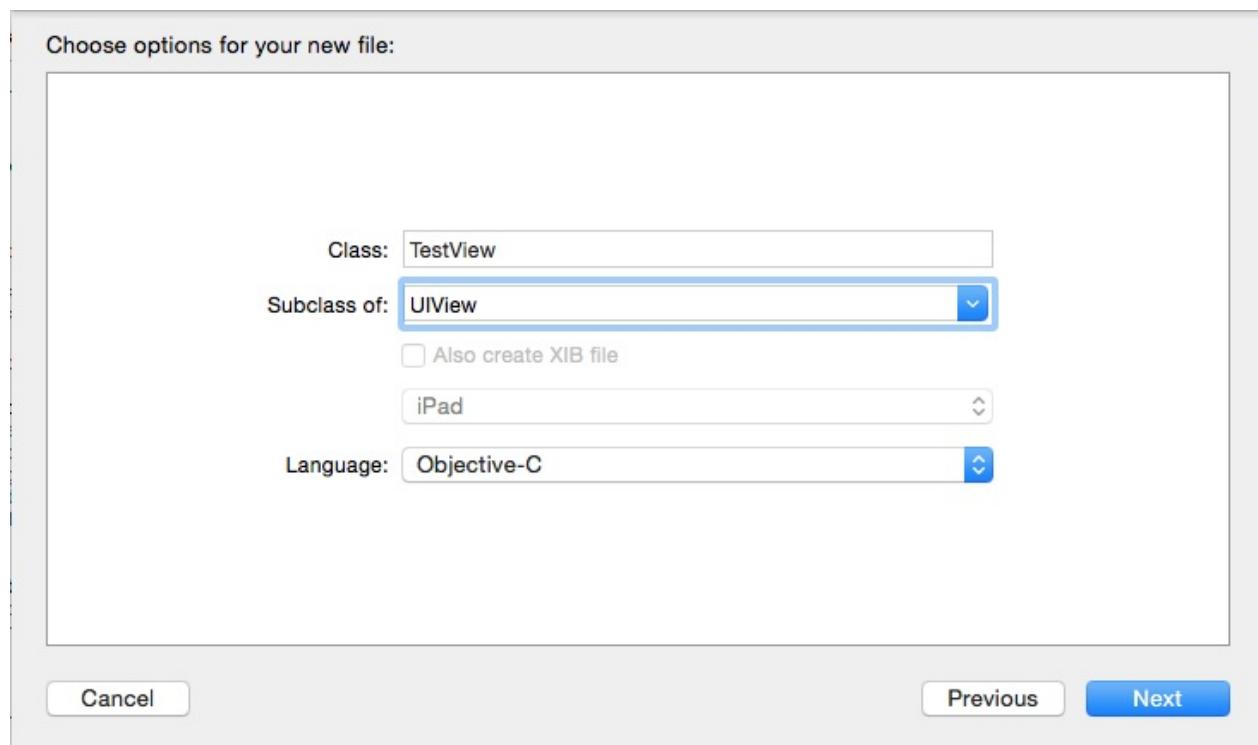
1. 需要实现较为复杂的 View 布局，通过 xib 方式实现

## 参考链接

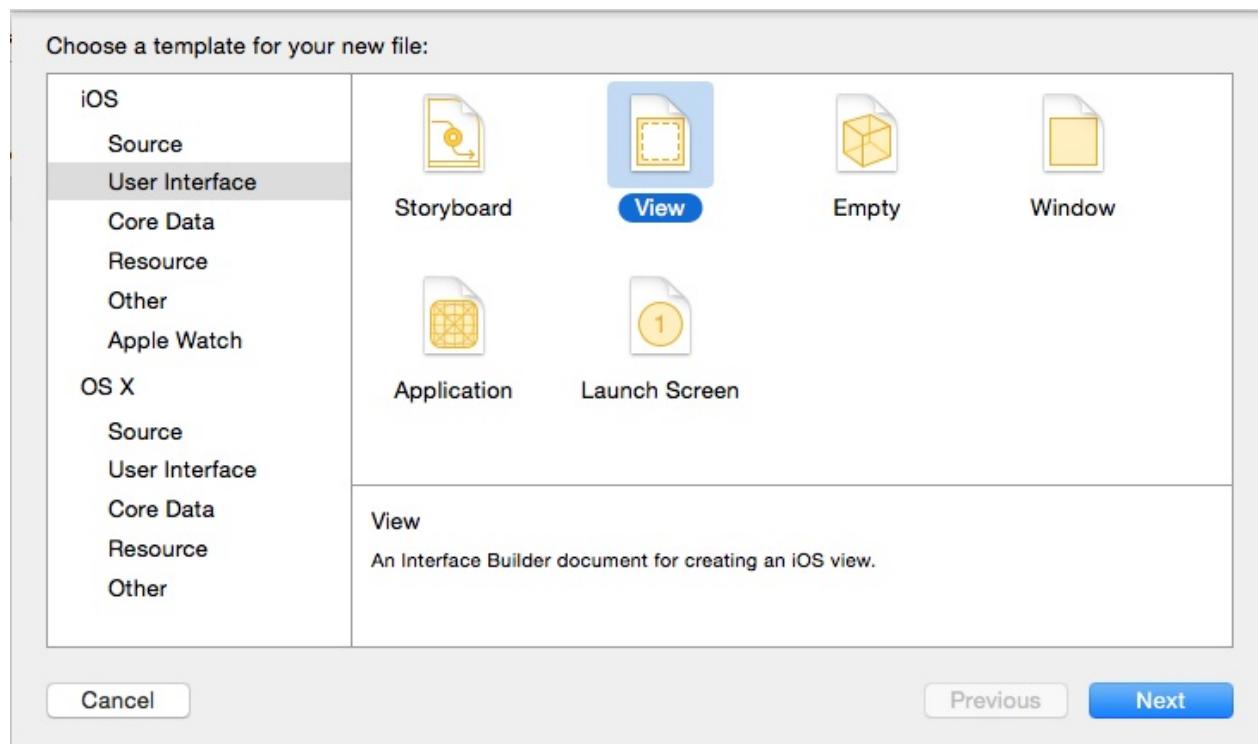
(无)

## 详细内容

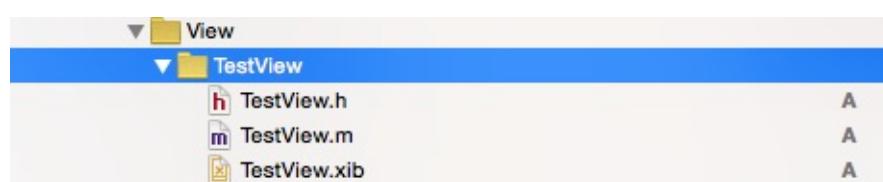
1. 通过 **File -> New -> File... -> iOS -> Source -> Cocoa Touch Class** 创建 View 的 **Class** 文件
2. 在 **Class** 中输入自定义 View 名称(示例：**TestView**)；在 **Subclass of** 中输入父类 **UIView**；此处无法勾选 **Also create Xib file** 选项（不用着急，往下看）；选择平台 (**iPhone / iPad**) 和语言 (**Objective-C / Swift**)



### 3. 通过 File -> New -> File... -> iOS -> User Interface -> View 创建 View 的 Xib 文件 (示例命名为 TestView)



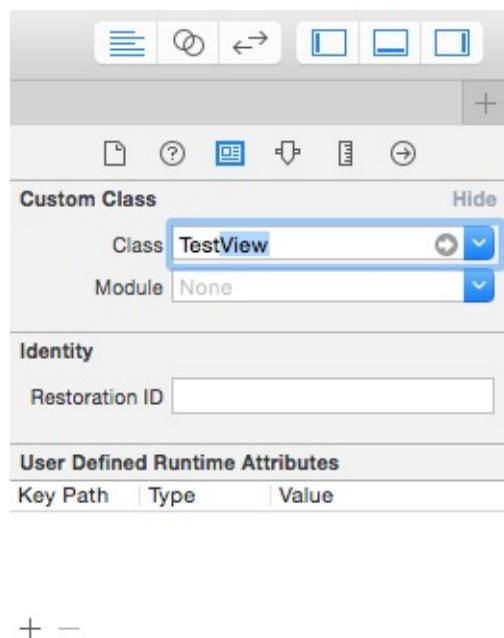
创建完成显示如下



**4. 下面进入 TestView.xib -> View -> Utilities 面板 -> Show the Identity inspector**

**5. 在 Class 中输入对应 View 的 Class 名称(示例 : TestView), 这样就实现了 Xib 和类文件的关联, 现在你可以像在 UIViewController 的 Xib 中拖动控件一样的在 View 上拖动关联控件了**

(注意, 此时 xib 与 Class 发生关联的是 View 而不是 File's Owner, 这与 UIViewController 和 xib 的关联方式不同)



**5. 在需要加载 xib 自定义 View 的地方使用如下代码加载**

```
NSArray *nib = [[NSBundle mainBundle] loadNibNamed:@"TestView" o
when:self options:nil];
TestView *tmpView = (TestView *)[nib objectAtIndex:0];
//这里的 0 表示 xib 中第 1 个View, 你可以尝试为一个 View 的 Class 添加
多个 View 控件
```

## 效果图

(无)

## 备注

1. View 是一个显示控件，尽量避免在 View 的 Class 中执行过多的逻辑运算和控制代码，View 应该通过 show 对应 Model 的方式显示自己的数据。

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-08 | Alfred Jiang | -  |

## 方案名称

专题 - iOS 使用 XMPP 实现即时聊天

## 关键字

专题 \ XMPP \ 即时聊天

## 需求场景

1. 利用 XMPP 实现即时聊天

## 参考链接

(见详细内容)

## 详细内容

1. [GitHub - IOSXMPPDemoByJackyHuang](#)
2. [标哥的技术博客 - iOS XMPP : Openfire+spark环境搭建（一）](#)
3. [标哥的技术博客 - iOS XMPP : 了解XMPP协议（二）](#)
4. [标哥的技术博客 - XMPPFramework核心类介绍（三）](#)
5. [标哥的技术博客 - iOS XMPP花名册细讲（四）](#)
6. [标哥的技术博客 - iOS XMPP : 电子名片细讲（五）](#)
7. [CSDN - IOS XMPP Study\(推荐\)](#)

## 其他

- [GitHub - nplexity/xmpp-file-transfer-demo](#)

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-19 | Alfred Jiang | -  |

## 方案名称

专题 - iOS 应用证书、真机调试与发布 Appstore

## 关键字

专题 \ 证书 \ 开发证书 \ 应用证书 \ 真机调试 \ Appstore

## 需求场景

1. 需要配置 iOS 开发证书相关

## 参考链接

(见详细内容)

## 详细内容

- iOS 证书学习推荐 - iOS证书及ipa包重签名探究
- iOS 证书学习推荐 - 代码签名探析
- 博客园 - In App Purchases 入门[译](原文)
- 博客园 - iOS 推送小结--swift语言
- 开源中国 - iOS Developer : 真机测试
- 开源中国 - iOS 发布应用程序到App Store
- 开源中国 - iOS 申请发布证书-图文详解

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |
| 2  | 2015-12-24 | Alfred Jiang | -  |

## 方案名称

专题 - iOS 开发学习资源参考

## 关键字

专题 \ iOS 开发 \ 入门与提高

## 需求场景

1. 需要学习 iOS 开发完整知识的参考专题

## 参考链接

(见详细内容)

## 详细内容

- A Better Way to Learn Swift
- Bloc—Build your first iOS game
- Cocoa Dev Central
- CocoaChina - iOS的设计备忘录/资源集合
- CSDN - iOS 技能图谱
- Developing iOS 8 Apps with Swift by Stanford
- GitHub - iOS Core Animation Advanced Techniques
- GitHub - iOS 学习资料整理 Aufree/trip-to-iOS
- Learn iOS SDK Development From Scratch

- NSCookbook
- Ry's Objective-C Tutorial
- Skyfox - iOS 开发学习路线图
- Swift Essentials - Learn Swift 2.1 Step by Step
- Tutorials Point - iOS (iPhone, iPad) Tutorial
- 慕课网 - 28岁，零基础，我是如何学习 iOS 开发的
- 知乎 - GitHub 上面有哪些适合新手阅读和学习的关于 iOS 开发的开源项目？

## 效果图

(无)

## 备注

- 专题 - iOS 开发学习资源参考
- 专题 - iOS 开发第三方代码库参考
- 专题 - 优秀技术博客参考
- 专题 - 优秀技术文章参考

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-17 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |
| 2  | 2015-12-24 | Alfred Jiang | -  |

## 方案名称

专题 - iOS 开发工具软件集合

## 关键字

专题 \ Mac 系统 \ MAC OS \ iOS \ 开发工具 \ 开发软件 \ 工具软件 \ 辅助工具

## 需求场景

1. 提高 iOS 开发效率

## 参考链接

1. [John Liu Thinks](#)
2. [CocoaChina - 【译】17个提升iOS开发效率的必用工具](#)
3. [17 Must Have Tools to Speed up Development in iOS](#)
4. [伯乐在线 - iOS 开发者必不可少的 75 个工具](#)

## 详细内容

| 类别  | 工具                          | 说明                |
|-----|-----------------------------|-------------------|
| App | <a href="#">Charles</a>     | 网络截包工具            |
| App | <a href="#">Cornerstone</a> | Svn               |
| App | <a href="#">iBackupbot</a>  | 查看非越狱真机 APP 目录下文件 |

| App          | <a href="#">ImageOptim</a>                | 图片压缩                                         |
|--------------|-------------------------------------------|----------------------------------------------|
| App          | <a href="#">Kaleidoscope</a>              | 文件对比工具 / Diff 工具（提供比 FileMerge 功能更强大的文件对比功能） |
| App          | <a href="#">Linguage</a>                  | 帮助提高本地化文件编辑效率                                |
| App          | <a href="#">MacDown</a>                   | Markdown 编辑工具                                |
| App          | <a href="#">MindNode</a>                  | 思维导图                                         |
| App          | <a href="#">Network Link Conditioner</a>  | 模拟网络环境                                       |
| App          | <a href="#">OmniGraffle</a>               | 绘图软件                                         |
| App          | <a href="#">SimPholders</a>               | 快速跳转至模拟器工程目录                                 |
| App          | <a href="#">SourceTree</a>                | Git                                          |
| App          | <a href="#">Sublime Text 3</a>            | 编辑器                                          |
| App          | <a href="#">Unused</a>                    | 检查工程中未使用的文件，帮助减小安装包大小                        |
| App          | <a href="#">Visual XML</a>                | API 测试工具                                     |
| App          | <a href="#">VisualJSON</a>                | API 测试工具                                     |
| Chrome 插件    | <a href="#">Rest Console</a>              | API 测试工具                                     |
| Firefox 插件   | <a href="#">SQLiteManager</a>             | 可视化 SQLite 数据库管理工具                           |
| Photoshop 插件 | <a href="#">Cutterman</a>                 | Photoshop 快速切图工具                             |
| Web 工具       | <a href="#">Appscreens</a>                | 免费在线生成 AppStore 审核用截图                        |
| Xcode 插件     | <a href="#">Alcatraz</a>                  | Xcode 插件管理工具                                 |
| Xcode 插件     | <a href="#">BBUncrustifyPlugin-Xcode</a>  | 代码格式化                                        |
| Xcode 插件     | <a href="#">ColorSense</a>                | 颜色辅助工具                                       |
| Xcode 插件     | <a href="#">Derived Data Exterminator</a> | 快速清理 Derived Data                            |
| Xcode 插件     | <a href="#">FuzzyAutocomplete</a>         | 自动补全代码                                       |

|          |                                                   |                          |
|----------|---------------------------------------------------|--------------------------|
| Xcode 插件 | <a href="#">HOStringSense</a>                     | 统计字符串字数以及实现字符转义          |
| Xcode 插件 | <a href="#">KSImageNamed</a>                      | 图片辅助工具                   |
| Xcode 插件 | <a href="#">ObjectGraph</a>                       | 简易类图导出工具                 |
| Xcode 插件 | <a href="#">QuickLocalization</a>                 | 帮助提高本地化文件编辑效率            |
| Xcode 插件 | <a href="#">SCXcodeSwitchExpander</a>             | 自动补全 Switch 分支           |
| Xcode 插件 | <a href="#">ShowInGitHub</a>                      | 查询 GitHub 代码             |
| Xcode 插件 | <a href="#">VVDocumenter-Xcode</a>                | 快速实现注释                   |
| Xcode 插件 | <a href="#">XAlign</a>                            | 自动补齐插件                   |
| Xcode 插件 | <a href="#">XcodeBoost</a>                        | 代码高亮、快速生成方法声明、正则查询、快速复制等 |
| 数据库      | <a href="#">GitHub - Commander147/SQLExplorer</a> | 可视化数据库管理工具               |
| 数据库      | <a href="#">Liya</a>                              | 可视化数据库管理工具               |
| 终端 工具    | <a href="#">Chisel</a>                            | LLDB 增强工具                |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

专题 - iOS 开发第三方代码库参考

## 关键字

专题 \ iOS 开发 \ 第三方库

## 需求场景

1. 需要第三方库快速实现业务需求

## 参考链接

(见详细内容)

## 详细内容

### 代码库

- [Github - iOS备忘](#)
- [Code4App](#)
- [CocoaChina 代码库](#)
- [OSChina 代码库](#)
- [DevDiv 代码库](#)

### 第三方库

- [27 iOS open source libraries to skyrocket your development.](#)

- [CocoaChina - 超全！整理常用的iOS第三方资源](#)

## 第三方组件

- [GitHub - CoreStore : Core Data 是 Apple 官方的数据关系图框架](#)
- [GitHub - LeeGo : LeeGo 是一个尝试使用声明式的，可配置的和易复用的 UI 组件进行用户界面开发的方式](#)
- [GitHub - Palau : 用类型安全的方式对 NSUserDefaults 进行了封装，同时也扩展了 NSUserDefaults 所能支持的类型，让我们可以用更好的方式使用 NSUserDefaults](#)
- [GitHub - SwiftOCR : SwiftOCR 是一个用 Swift 写的 OCR 框架](#)
- [GitHub - YYKit : A collection of iOS components.](#)

## 效果图

(无)

## 备注

- [专题 - iOS 开发学习资源参考](#)
- [专题 - iOS 开发第三方代码库参考](#)
- [专题 - 优秀技术博客参考](#)
- [专题 - 优秀技术文章参考](#)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-07-22 | Alfred Jiang | -  |

## 方案名称

专题 - iOS 开发调试代码集合

## 关键字

专题 \ iOS \ 开发调试代码

## 需求场景

1. 提高 iOS 开发调试效率

## 参考链接

(无)

## 详细内容

1. 测试文件读写

```

        NSArray *paths=NSSearchPathForDirectoriesInDomains(NSDo
cumentDirectory
        , NS
UserDomainMask
        , YES
);
 NSLog(@"Get document path: %@",[paths objectAtIndex:0])
;

NSString *filePath = [[paths objectAtIndex:0] stringByAppendingA
ppendingPathComponent:@"testFile"];

if ([NSKeyedArchiver archiveRootObject:userInfo toFile:
filePath]) {
    NSLog(@"Save Successed");
}

```

## 2. 延时执行

```

//Block
void RunBlockAfterDelay(NSTimeInterval delay, void (^block)(void
))
{
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, NSEC_PER_SEC
* delay), dispatch_get_main_queue(), block);
}

//直接等待
[NSThread sleepForTimeInterval:7.0f];

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-05-06 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

专题 - iOS 设备处理器指令集参考

## 关键字

专题 \ 处理器 \ 指令集 \ Architectures

## 需求场景

1. 需要对所编译文件考虑指令集设置需求时
2. 编译静态库需求

## 参考链接

1. [CocoaChina - Apple移动设备处理器指令集 armv6、armv7、armv7s及arm64](#)

## 详细内容

### **armv6** 设备：

- iPhone, iPhone2, iPhone3G
- 第一代、第二代 iPod Touch

### **armv7** 设备：

- iPhone3GS, iPhone4, iPhone4S
- iPad, iPad2, iPad3(The New iPad), iPad mini
- iPod Touch 3G, iPod Touch4

**armv7s**设备：

- iPhone5, iPhone5C
- iPad4(iPad with Retina Display)

**arm64**设备：

- iPhone6 Plus, iPhone6, Phone5S
- iPad Air, iPad mini2(iPad mini with Retina Display)

更多指令集设置相关可参考 [《Apple移动设备处理器指令集 armv6、armv7、armv7s及arm64》](#)

**效果图**

(无)

**备注**

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注            |
|----|------------|--------------|---------------|
| 1  | 2015-12-28 | Alfred Jiang | -             |
| 2  | 2016-02-02 | Alfred Jiang | -             |
| 3  | 2016-02-14 | Alfred Jiang | 更新 Mac 开源项目相关 |

## 方案名称

专题 - Mac & iOS 完整开源项目源代码参考

## 关键字

专题 \ Mac \ iOS \ 完整 \ 项目 \ 工程 \ 源代码

## 需求场景

- 希望通过完整的项目工程源代码学习 Mac \ iOS 开发。

## 参考链接

- 知乎 - GitHub 上有哪些完整的 iOS-App 源码值得参考
- OPEN SOURCE IPHONE SOFTWARE - iOS 平台上一些比较有名的开源软件集合(推荐阅读)
- OPEN SOURCE MAC - Mac OS X 平台上一些比较有名的开源软件集合(推荐阅读)
- 标哥的技术博客 - iOS完整App资源收集
- GitHub - dkhamsing/open-source-ios-apps
- GitHub - zJunZ/CollectGithubAPP

## 详细内容

### iOS



| 名称                | 语言<br>(>10%)        | 简介                                                                                                       |
|-------------------|---------------------|----------------------------------------------------------------------------------------------------------|
| Awesome-Apple     | Markdown            | A list of awesome 3rd party libraries and tools for Apple platform development, iOS and Mac.             |
| Awesome-iOS       | Swift               | A curated list of awesome iOS frameworks, libraries, tutorials, Xcode plugins, components and much more. |
| Awesome-Swift     | Swift               | An awesome list of Swift related guides.                                                                 |
| BTCChinaTrade     | Objective-C         | 比特币中国 • iOS 交易客户端                                                                                        |
| Coding-iOS        | Objective-C         | Coding-iOS 是 @Coding 官方开源的一个完整的 iOS 应用，里面的很多代码结构，模块功能划分，第三方服务的集成使用等经验都非常值得开发者学习和参考                       |
| EarlGrey          | Objective-C         | 由 Objective-C 编写而成的用户界面测试工具                                                                              |
| Firefox for iOS   | C / Swift           | 火狐浏览器 iOS 版                                                                                              |
| GitBucket         | Objective-C         | 基于MVVM和RAC的GitHub第三方开源客户端                                                                                |
| HappyEnglish      | Objective-C         | 一款英语学习 App                                                                                               |
| Inkpad            | Objective-C / C++   | iOS / iPad 一个功能十分完整、效果很不错的矢量图绘制App源代码                                                                    |
| Monkey for GitHub | Objective-C         | 一个 GitHub 第三方客户端                                                                                         |
| PHPHub for iOS    | Objective-C         | PHPHub for iOS 是 PHPHub 的官方 iOS 客户端，同时兼容 iPhone 和 iPad，使用了最新的 PHPHub API                                 |
| RSSRead           | Objective-C / HTML  | “已阅”是一个iOS设备上的RSS/Atom阅读器                                                                                |
| WhoCall           | Objective-C         | iOS来电信息语音提醒，无需越狱（需要iOS 7.0及以上版本）                                                                         |
| Wikipedia iOS     | Objective-C / HTML  | The official Wikipedia iOS client                                                                        |
| WordPress for iOS | Objective-C / Swift | WordPress iOS 客户端                                                                                        |

Yep

Swift

Yep 是一款非常小巧而轻量化的社交 App

## Mac

| 名称                                   | 语言<br>(>10%)                   | 简介                                                                                                                                         |
|--------------------------------------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Core-Data-Editor                     | Objective-C                    | Core Data Editor lets you easily view, edit and analyze applications' data                                                                 |
| Launchpad-Control                    | Objective-C                    | <a href="#">Launchpad-Control</a> is a small tool which allows you to easily hide/unhide apps (and groups) from Launchpad in Mac OS X Lion |
| MacDown                              | Objective-C / C                | MacDown is an open source Markdown editor for OS X                                                                                         |
| Markoff                              | Swift / CSS / JavaScript       | Mac 上面轻量级的 Markdown 文件预览工具                                                                                                                 |
| MeshMaker                            | C++                            | Mac 下多边形建模工具                                                                                                                               |
| Poison                               | Objective-C / C                | Poison is a Mac client for <a href="#">Tox</a>                                                                                             |
| Radiant Player for Google Play Music | Objective-C / JavaScript / CSS | Google 音乐 Mac 版                                                                                                                            |
| RegExhibit                           | Objective-C                    | RegExhibit is a program that allows you to interactively examine and test regular expressions                                              |
| Unused                               | Objective-C                    | 检查工程中未使用的文件，帮助减小安装包大小                                                                                                                      |
| WWDC                                 | Swift / Objective-C            | WWDC app for OS X                                                                                                                          |

更多详细内容见参考链接-推荐阅读

## 效果图

(无)

## 备注

- 老谭笔记 : 中文 Mac 开发博客

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-09-30 | Alfred Jiang | -  |

## 方案名称

专题 - objc.io Issue 3: 视图 相关文章五篇

## 关键字

专题 \ Views \ 视图

## 需求场景

- 熟悉 iOS 视图层相关编程

## 参考链接

- objc中国 - 期刊(推荐)

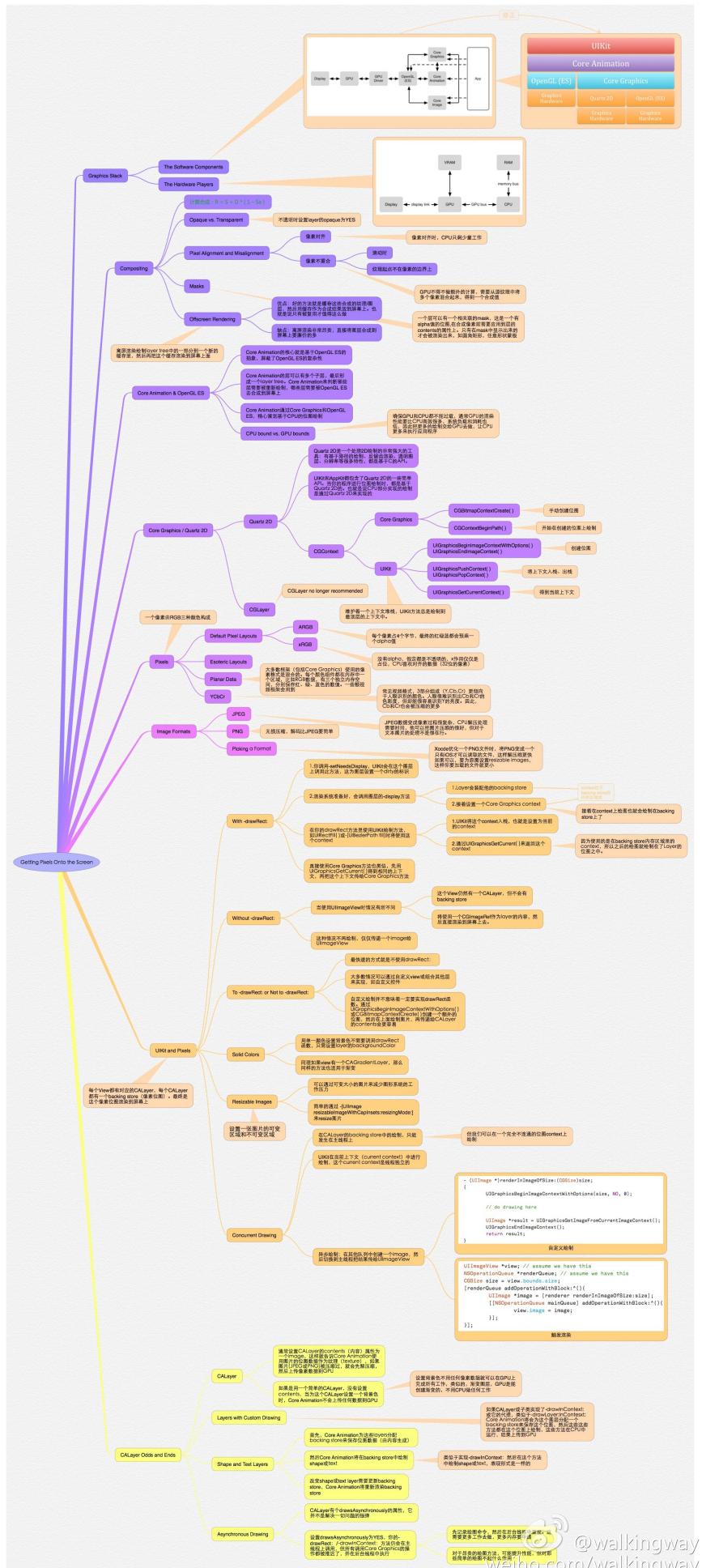
## 详细内容

结合 walkingway 新浪微博提供的五张高清图，熟悉《objc.io Issue 3: 视图》期刊五篇关于 iOS 视图层编程的介绍文章。以下为objc中国 - 期刊中文翻译链接

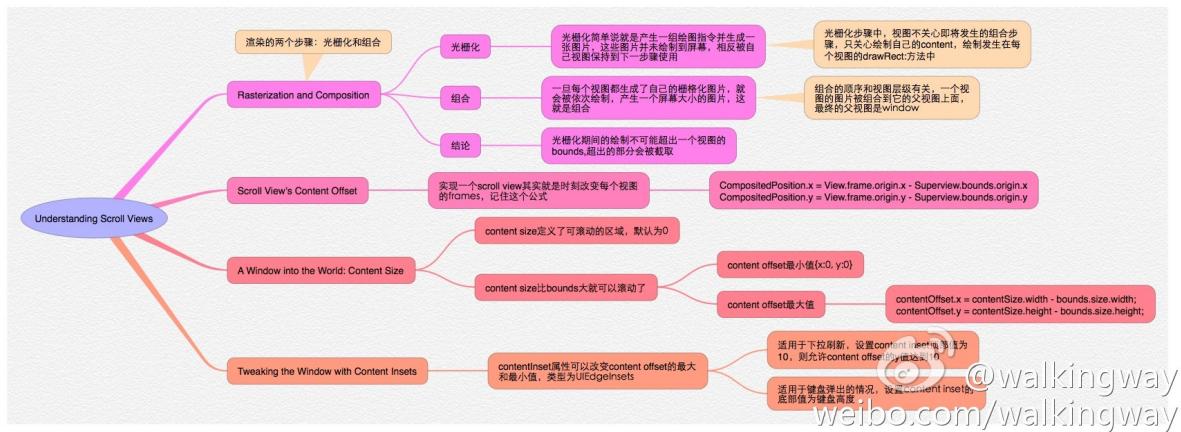
- 绘制像素到屏幕上
- 理解 Scroll Views
- 自定义 Collection View 布局
- 自定义控件
- 先进的自动布局工具箱

## 效果图

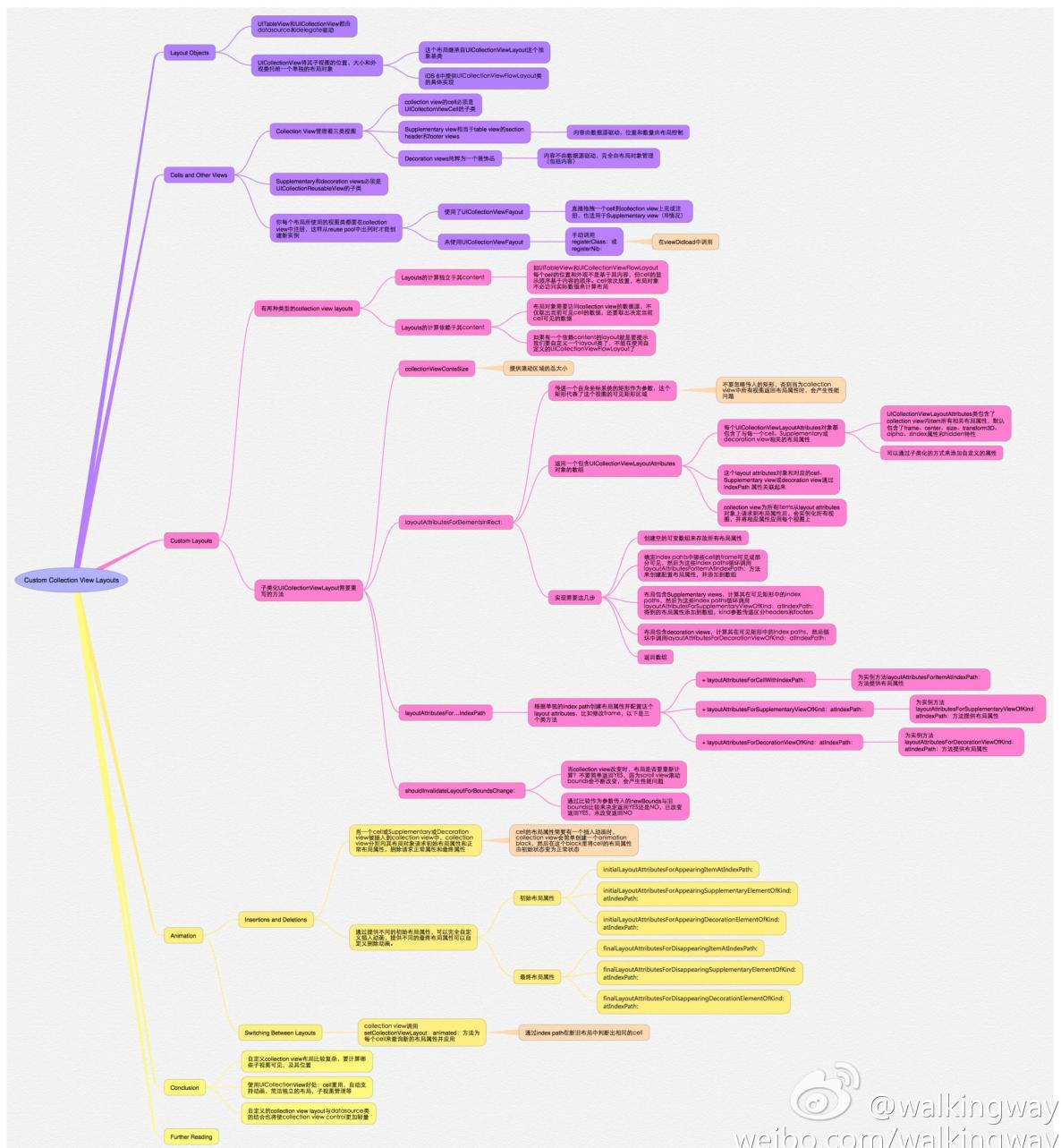
- 绘制像素到屏幕上



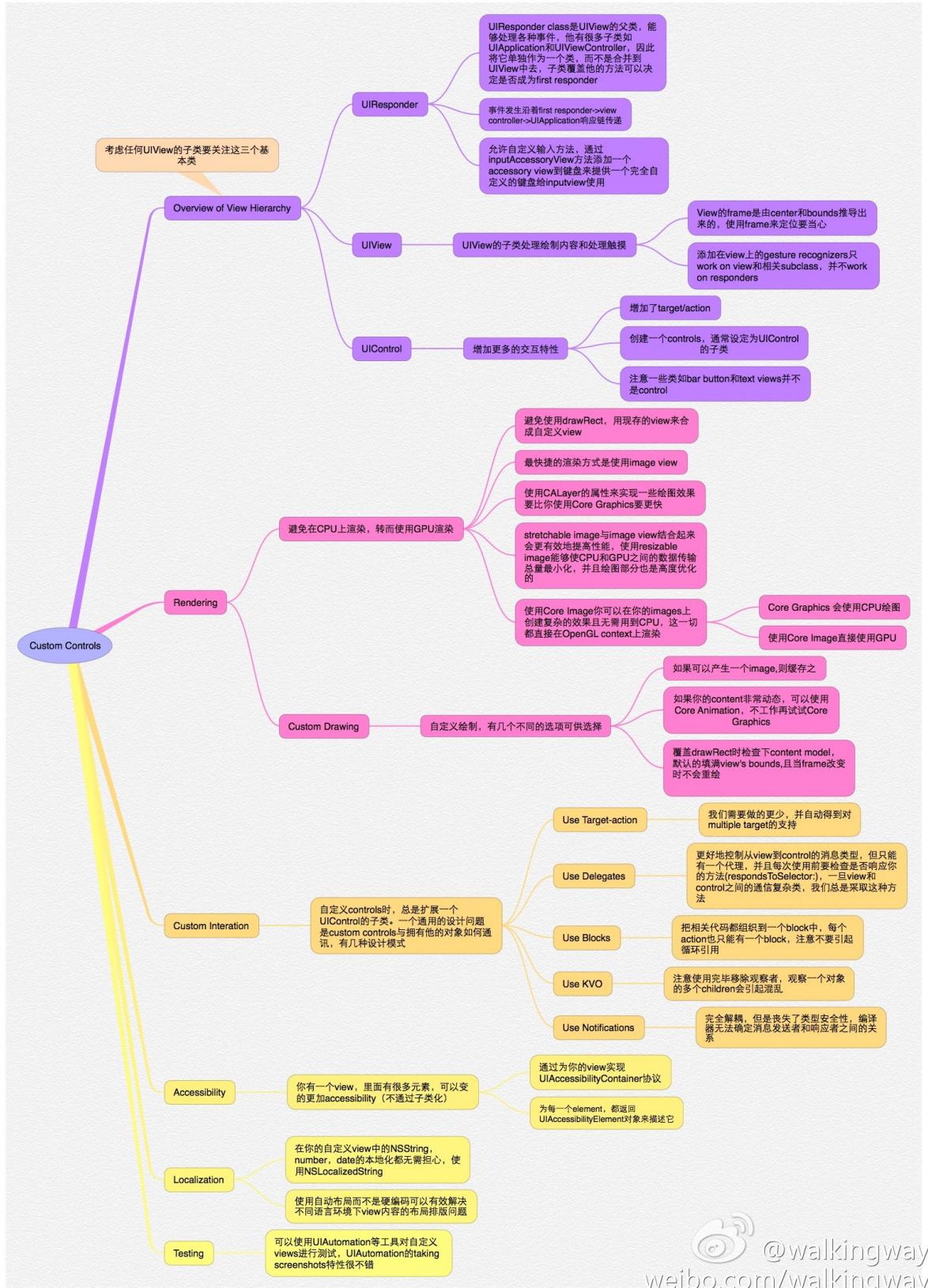
## ● 理解 Scroll Views



## ● 自定义 Collection View 布局

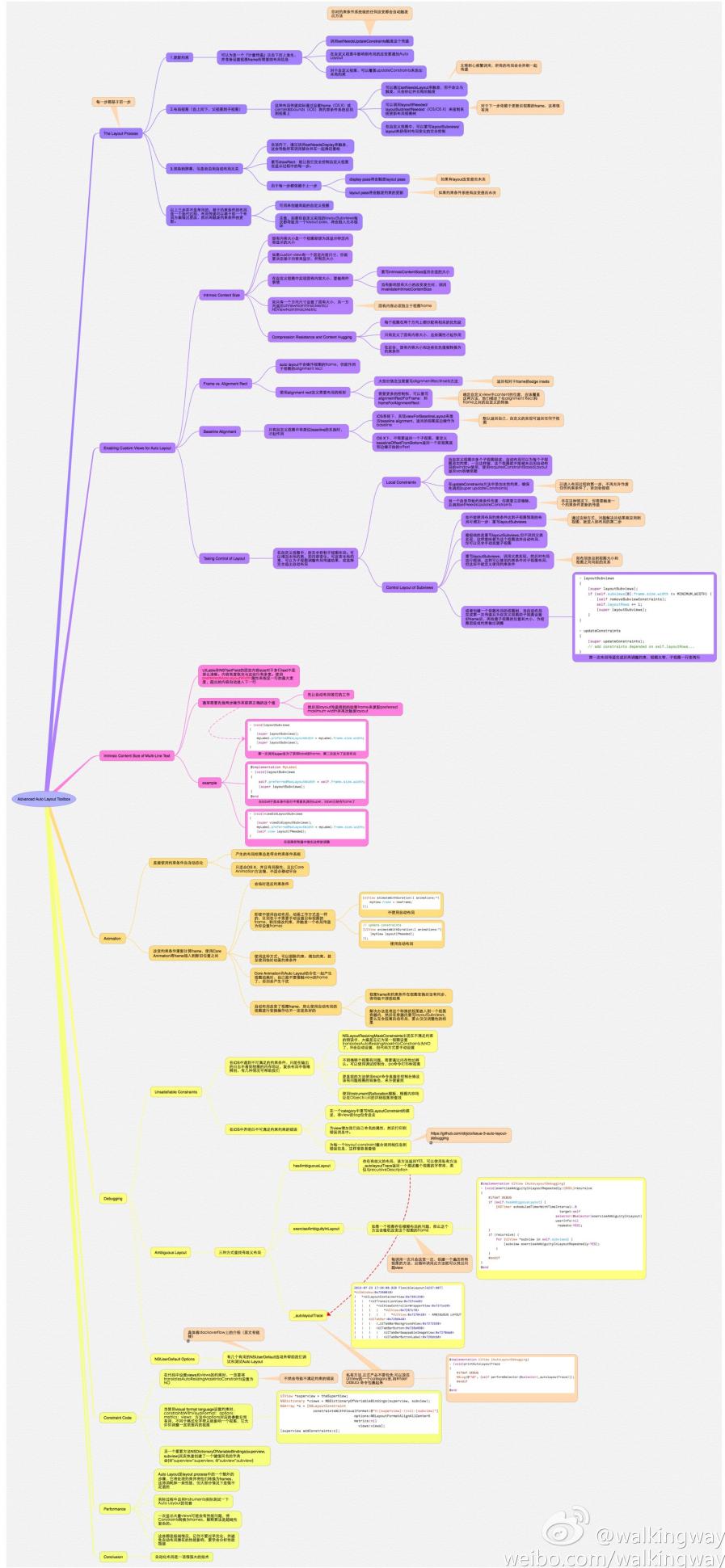


## ● 自定义控件



@walkingway  
weibo.com/walkingway

## ● 先进的自动布局工具箱



## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-20 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |
| 2  | 2015-12-24 | Alfred Jiang | -  |

## 方案名称

专题 - 优秀技术博客参考

## 关键字

专题 \ 参考 \ 博客 \ 文章 \ 技术

## 需求场景

1. 用于收集优秀的技术博客参考

## 参考链接

(见详细内容)

## 详细内容

iOS 开发相关网站资源

- App Coder
- AppCoda
- Apple documentation
- Apple documentation—Swift Overview
- Bartosz Ciechanowski
- Big Nerd Ranch Blog

- [Cocoa at Tumblr](#)
- [Cocoa Is My Girlfriend](#)
- [Coding Explorer](#)
- [Conrad Stoll](#)
- [Florian Kugler](#)
- [iCode blog](#)
- [iOS Dev Weekly](#)
- [iOS Developer Tips](#)
- [iOS Development tips](#)
- [iOS Goodies](#)
- [iOS-Blog](#)
- [Krzysztof Zab?ocki](#)
- [ManiacDev](#)
- [Natasha The Robot](#)
- [Nils Hayat](#)
- [NSBlog](#)
- [NSHint](#)
- [NSHipster](#)
- [NSHipster中文版](#)
- [NSScreencast](#)
- [objc.io](#)
- [objc中国](#)
- [Ole Begemann](#)
- [Peter Steinberger](#)
- [Ray Wenderlich](#)
- [Subjective-C](#)
- [Use Your Loaf](#)
- [唐巧的技术博客](#)
- [标哥的技术博客](#)
- [破船之家](#)
- [Hamster Emporium](#)

其他一

- [ASCIIwwdc](#)
- [GitHub](#)

- Stack Overflow

其他二

- 11 Insanely Great iOS Developers Sites
- iOS优秀博客收录（持续更新）
- IOS编程开发
- 取精华、去糟粕！适合iOS开发者的15大网站推荐

效果图

(无)

备注

- 专题 - iOS 开发学习资源参考
- 专题 - iOS 开发第三方代码库参考
- 专题 - 优秀技术博客参考
- 专题 - 优秀技术文章参考

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-07-15 | Alfred Jiang | -  |
| 2  | 2015-12-20 | Alfred Jiang | -  |
| 2  | 2015-12-24 | Alfred Jiang | -  |

## 方案名称

专题 - 优秀技术文章参考

## 关键字

专题 \ 参考 \ 博客 \ 文章 \ 技术

## 需求场景

1. 用于收集优秀的技术文章参考

## 参考链接

(见详细内容)

## 详细内容

iOS 代码规范

- [Avoiding Complex View Controllers](#)
- [iOS应用开发最佳实践系列一：编写高质量的Objective-C代码](#)
- [iOS应用性能调优的25个建议和技巧\[译\]\(原文\)](#)
- [Objective-C 的 API 设计\[译\]\(原文\)](#)

二维码相关

- [二维码的生成细节和原理](#)

## 设计类

- [iOS设计指南\[译\] \(原文\)](#)
- [wwwwhere 设计师常用链接](#)
- [苹果官方设计指南](#)
- [博客园 - 好的用户界面-界面设计的一些技巧](#)
- [慕课网 - 提高工作效率的10个UX & UI工具](#)

## SDK 框架介绍

- [Segmentfault - 传感器框架 CMDeviceMotion\[译\]\(原文\)](#)

## 其他一

- [27 iOS open source libraries to skyrocket your development](#)
- [27 places to learn iOS development. Best ones](#)
- [iOS 自定义控件资源大全](#)
- [IOS开发之----资源汇总](#)
- [iOS开发如何提高](#)
- [Projects that power GitHub for Mac](#)

## 其他二

- [\(器\) 构建自由通行的IOS开发者地图](#)
- [\(术\) IOS应用的产品设计、交互设计理念断想](#)
- [\(法\) 平衡出最优的team生产力组合](#)

## 其他三

- [绝密原型档案：看看专业产品经理的原型是什麼样](#)

## 其他四

- [How To Configure Secure Updates and Installations in WordPress on Ubuntu](#)

## 第三方库源码解析

- [简书 - SDWebImage源码解析（一）](#)
- [简书 - SDWebImage源码解析（二）](#)
- [简书 - SDWebImage源码解析（三）](#)

## 面试类

- 一个多月来的面试总结(阿里, 网易, 腾讯)
- IOS面试问题总结

## 效果图

(无)

## 备注

- 专题 - iOS 开发学习资源参考
- 专题 - iOS 开发第三方代码库参考
- 专题 - 优秀技术博客参考
- 专题 - 优秀技术文章参考

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-20 | Alfred Jiang | -  |

## 方案名称

专题 - 值得模块化的业务需求

## 关键字

专题\模块化\业务需求

## 需求场景

- 可以模块化的业务需求参考

## 参考链接

(无)

## 详细内容

| 序号 | 名称          | 备注                               |
|----|-------------|----------------------------------|
| 1  | 登陆模块        | -                                |
| 2  | 注册模块        | -                                |
| 3  | 地图标签与路线绘制模块 | -                                |
| 3  | 网络请求模块      | -                                |
| 3  | 即时聊天模块      | socket.io、XMPP、MessageDisplayKit |
| 3  | 数据库管理模块     | -                                |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-24 | Alfred Jiang | -  |

## 方案名称

专题 - 创建自己的 iOS 框架

## 关键字

专题 \ iOS 框架 \ framework

## 需求场景

1. 为其他项目提供 framework 框架时

## 参考链接

1. [Creating your first iOS Framework](#)
2. [Segmentfault - 创建你的第一个iOS框架\(推荐\)](#)
3. [GitHub - 王巍\\_如何打造一个让人愉快的框架\(推荐\)](#)

## 详细内容

(见参考链接)

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-24 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

专题 - 可视控件介绍与可定制替代方案推荐

## 关键字

专题 \ 系统控件 \ 控件 \ UIKit \ 自定义系统控件

## 需求场景

1. 需要对 SDK 所提供系统控件梳理了解时
2. 需要自定义系统控件时

## 参考链接

1. [Code4App](#)
2. [Apple documentation](#)

## 详细内容

### iPhone 与 iPad 共有可视控件

| 序号 | 控件   | Class  | 替代增强方案 | 说明   |
|----|------|--------|--------|------|
| 1  | View | UIView |        | 视图控件 |
|    |      |        |        | 搜    |

|   |                 |                                  |  |           |
|---|-----------------|----------------------------------|--|-----------|
| 2 | Search Bar      | <a href="#">UISearchBar</a>      |  | 索控件       |
| 3 | Tab Bar         | <a href="#">UITabBar</a>         |  | 底部标签控件    |
| 4 | Tab Bar Item    | <a href="#">UITabBarItem</a>     |  | 底部标签控件按钮项 |
| 5 | Toolbar         | <a href="#">UIToolbar</a>        |  | 底部工具栏控件   |
| 6 | Navigation Bar  | <a href="#">UINavigationBar</a>  |  | 导航控件      |
| 7 | Navigation Item | <a href="#">UINavigationItem</a> |  | 顶部导航控件按钮项 |
|   |                 |                                  |  | 底部工具栏控    |

|    |                 |                 |                        |                |
|----|-----------------|-----------------|------------------------|----------------|
|    |                 |                 |                        | 件按钮项或顶部导航控件按钮项 |
| 8  | Bar Button Item | UIBarButtonItem | UIBarButtonItem-Badge  |                |
| 9  | Web View        | UIWebView       |                        | 网页控件           |
| 10 | iAd BannerView  | ADBannerView    |                        | iAd 广告控件       |
| 11 | MapKit View     | MKMapView       |                        | 地图控件           |
| 12 | Picker View     | UIPickerView    |                        | 选择器控件          |
| 13 | Date Picker     | UIDatePicker    | MGConferenceDatePicker | 日期选择器控件        |
| 14 | Scroll View     | UIScrollView    |                        | 滑动视图控          |

|    |                 |                  |               | 件           |
|----|-----------------|------------------|---------------|-------------|
| 15 | Text View       | UITextView       | CYRTextView   | 段落文本编辑与显示控件 |
| 16 | Collection View | UICollectionView |               | 集合视图控件      |
| 17 | Image View      | UIImageView      | SDWebImage    | 图片显示控件      |
| 18 | Table View      | UITableView      |               | 列表现实控件      |
| 19 | Stepper         | UIStepper        |               | 增减计数控件      |
| 20 | Page Control    | UIPageControl    | SMPageControl | 页码展示控件      |
|    |                 |                  |               | 进度          |

|    |                         |                                         |                                       |        |
|----|-------------------------|-----------------------------------------|---------------------------------------|--------|
| 21 | Progress View           | <a href="#">UIProgressView</a>          | <a href="#">MBProgressHUD</a>         | 展示控件   |
| 22 | Activity Indicator View | <a href="#">UIActivityIndicatorView</a> | <a href="#">MBProgressHUD</a>         | 旋转等待控件 |
| 23 | Switch                  | <a href="#">UISwitch</a>                | <a href="#">KLSwitch</a>              | 开关控件   |
| 24 | Slider                  | <a href="#">UISlider</a>                | <a href="#">ASValueTrackingSlider</a> | 滑动条控件  |
| 25 | Text Field              | <a href="#">UITextField</a>             | <a href="#">MaterialKit</a>           | 文本输入控件 |
| 26 | Segmented Control       | <a href="#">UISegmentedControl</a>      |                                       | 分段控制控件 |
| 27 | Button                  | <a href="#">UIButton</a>                | <a href="#">MaterialKit</a>           | 按钮控件   |
| 28 | Label                   | <a href="#">UILabel</a>                 | <a href="#">MaterialKit</a>           | 文本显示控件 |
|    |                         |                                         |                                       | 弹出     |

|    |              |                               |                                    |           |
|----|--------------|-------------------------------|------------------------------------|-----------|
| 29 | Alert View   | <a href="#">UIAlertView</a>   | <a href="#">UIAlertView-Blocks</a> | 提示框控件     |
| 30 | Action Sheet | <a href="#">UIActionSheet</a> | <a href="#">AHKActionSheet</a>     | 底部弹出选择框控件 |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-05-11 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

专题 - 后台模式开发指南

## 关键字

专题 \ 音频播放 \ 接收位置更新 \ 执行有限长任务等 \ 后台获取

## 需求场景

1. 需要实现 iOS 后台需求时

## 参考链接

1. [GitHub - iOS后台模式开发指南](#)

## 详细内容

---

## iOS后台模式开发指南

- 原文链接：[Background Modes Tutorial: Getting Started](#)
- 原文作者：[Ray Fix](#)
- 译者：[MollyMmm](#)
- 校对者：[David Hu](#)
- 状态：已完成

更新说明:这个教程被Ray Fix更新为关于iOS和Swift的.原著作者为[Gustavo Ambrozio](#).

自从古老的iOS4以来,当用户点击home键的时候,你可以使你的APP们在内存中处于suspended(挂起)状态.即使APP仍停留在内存中,它的所有操作是被暂停的直到用户再次运行它.

当然这个规则中有例外情况.在特定的情况下,这个APP仍然可以在后台中执行某些操作.这个教程会教你在什么时候怎么去用最常用的一些后台操作.

每一次iOS的发布都会在后台操作和细节上的放宽限制,以此提升用户体验和延长电池寿命.对于在iOS中实现"真正"的多任务来说,后台模式不是一个神奇的解决办法.当用户切换到其他的APP应用时,大多数的APP应用仍然会完全的暂停运行.你的应用只被允许在很特殊的情况下才能在后台中继续运行.例如,这些包括播放音频,获取位置更新,或者从服务器获取最新内容的情况.

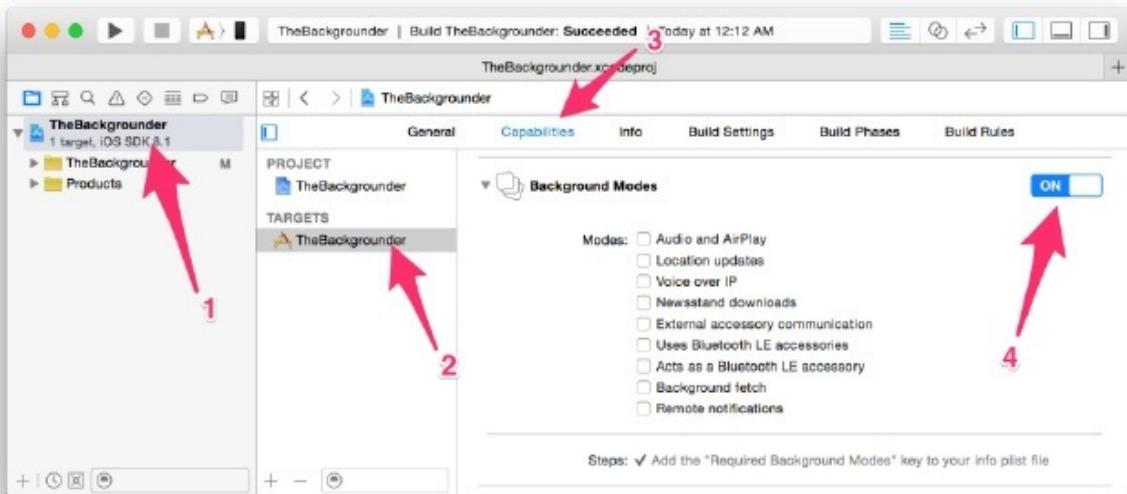
iOS7之前,APP应用在真正暂停之前会有连续10分钟的时间去完成它们当前的操作.随着NSURLSession的出现,有了一种更为优雅的方式去应对大量的网络切换.因此,对于可用的后台运行时间已经减少到只有几分钟,而且不再必须为连续的.

这样的后台模式可能不适合你.但如果合适,请继续阅读!

接下来的学习中,将会有几个几个后台模式提供给你.在本教程中你将建立一个关于简单标签应用的工程,来探索从连续播放视频到周期性的获取更新内容的四种常见模式.

## 开始

在深入这个工程之前,这里有一个iOS可用的基础后台模式的快速预览.在Xcode 6中,你通过点击目标程序的**Capabilities(功能)**选项卡能够看到如下列表:



打开后台模式功能列表(1)在项目导航栏中选择项目(2)选择目标应用(3)选择功能选项卡(4)把后台模式开关打开.

在这个教程中,你会研究四种后台进程处理方式.

\***视频播放:**APP可以在后台播放或录制视频

\***获取位置更新:**该应用会随着设备位置的改变继续回调结果.

\***执行一定的任务:**通常在没有限制的情况下,这时APP会在有限的时间内运行任意的代码.

\***后台获取:**通过iOS的更新计划获取最细的内容.

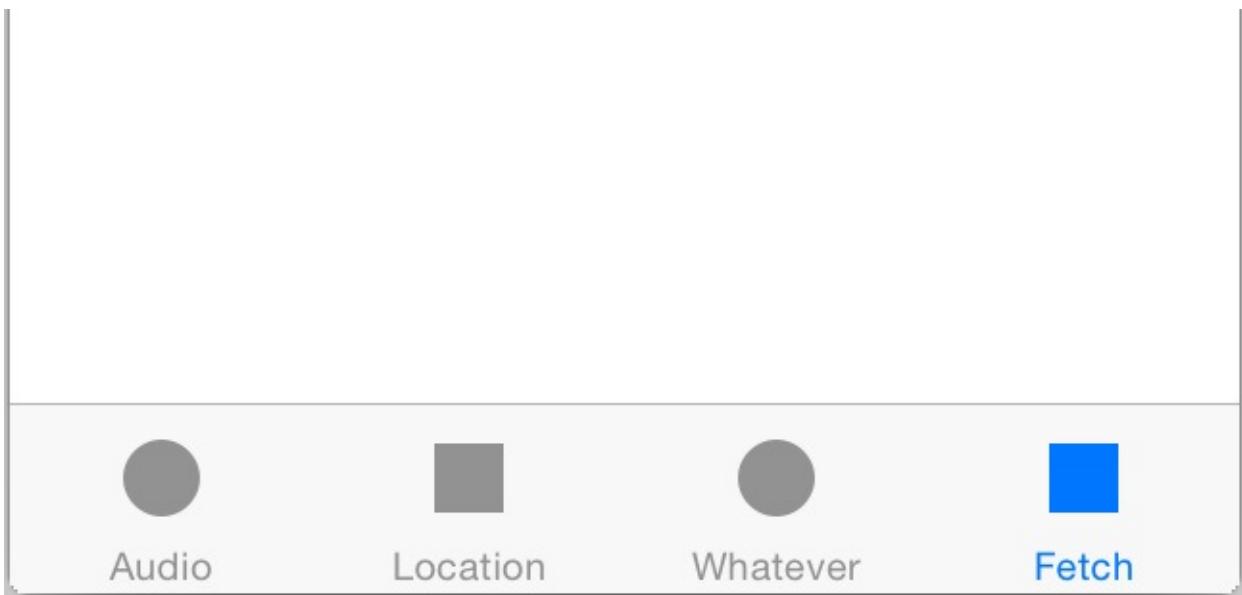
这个教程将按照上面的顺序,在本教程的每个部分中介绍如何使用这四个模板.

从这个像观光车一样的工程开始，通过它熟悉一下iOS后台机制，首先下载这个上手工程。有个好消息：用户界面已经为你预配置好了。



**Let's keep  
backgrounding in  
the foreground!**

运行这个示例项目，检查一下你的四个选项卡。



这些选项卡是本教程剩余部分的路线图。第一站：后台视频

提示：为了使后台模式充分发挥作用，你应该使用一个真正的设备。根据我的经验，如果你忘记配置设置，该APP在模拟器的后台能很好的运行。然而，当你切换到真正的设备时，它将不会运行。

## 音频播放

这里有iOS播放音频的几种方法，他们中的大部分需要实现回调函数去提供更多用来播放的音频数据。当用户使你的APP做某些事情，会调用回调函数（比如委托模型），在这种情况下，会把音波存储在内存缓存区中。

如果你想播放流数据中的音频,你可以开启一个网络连接,连接的这些回调函数提供连续的音频数据.

当你激活音频后台模式后,即使你的APP现在没在活动,iOS将继续执行这些回调函数.音频后台模式是自动的,这么说很正确.你只是激活它,恰好为管理它提供了基础设备.

对于我们这些有点小心思的人来说,如果你的APP确实为用户播放音频,你应该只使用后台音频模式.如果你尝试使用这个模式只是为了获取当程序安静运行的时候使用CPU的时长,苹果将拒绝你APP的运行.



在这部分,你将在你的APP中添加一个音频播放器,打开后台模式,为你演示它的运行过程.

为了获取到音频播放装置,你需要学习 AV Foundation. 打开 `AudioViewController.swift`, 在文件顶部 `import UIKit` 后面添加引用.

```
import AVFoundation
```

Override `viewDidLoad()` with the following implementation: 用下面的实现代码重写 `viewDidLoad()`

```

override func viewDidLoad() {
super.viewDidLoad()

var error: NSError?
var success = AVAudioSession.sharedInstance().setCategory(
AVAudioSessionCategoryPlayAndRecord,
withOptions: .DefaultToSpeaker, error: &error)
if !success {
NSLog("Failed to set audio session category. Error: \(error)")
}
}

```

这使用了音频回话的单例模式`sharedInstance()`去设置播放的类别,也确保了声音是通过手机扬声器而不是通过手机听筒传播的.如果它执行了,他会检查调用是否失败并记录错误.一个真正的APP在发生错误后会显示一个队伍的对话框,作为对错误的回应,但是我们不需要因为这些小细节而纠结.

接下来,你要把播放器这个成员属性添加到`AudioViewController`中:

```
var player: AVQueuePlayer!
```

这是个隐式的可拓展的属性,最初为`nil`,你将在`viewDidLoad()`对它进行初始化.

这个上手项目包含来自主要收纳免版权税的音乐网站incompetech.com的音频文件.认证之后你可以免费的使用它上面的音乐.你这里使用的全部歌曲来自incompetech.com 上Kevin MacLeod的作品.谢谢Kevin!

返回`viewDidLoad()`,在此函数的末尾处添加如下方法:

```

let songNames = ["FeelinGood", "IronBacon", "WhatYouWant"]
let songs = songNames.map {
AVPlayerItem(URL: NSBundle mainBundle().URLForResource($0, withExtension: "mp3"))
}

player = AVQueuePlayer(items: songs)
player.actionAtItemEnd = .Advance

```

这样可以获取到歌曲的列表,把它们映射到主程序包的路径中并把它们转化为可以在AVQueuePlayer上播放的AVPlayerItems.此外,这个队列被设置为循环播放.

为了在队列进程中更新歌曲名字,你需要观察播放器中的currentItem.为了达到上述目的,需要在viewDidLoad()的末尾处添加如下代码:

```
player.addObserver(self, forKeyPath: "currentItem", options: .New
    | .Initial, context: nil)
```

这使得每当播放器中currentItem改变,类观察者的回调被初始化.

现在你可以添加观察者模式方法.把下面代码放到viewDidLoad()下面.

```
override func observeValueForKeyPath(keyPath: String, ofObject o
bject: AnyObject, change: [NSObject : AnyObject], context: Unsafe
MutablePointer<Void>) {
    if keyPath == "currentItem", let player = object as? AVPlayer,
    currentItem = player.currentItem?.asset as? AVURLAsset {
        songLabel.text = currentItem.URL?.lastPathComponent ?? "Unknown"
    }
}
```

当这个函数被调用的时候,你首先要确保这个被更新的属性是你所关注的.在这种情形下,它不是那么重要了因为只有一个属性被观察,但是在你之后添加更多的观察者的情况下检查,是个不错的方法.如果它是currentItem键,你将使用它通过文件名更新songLabel.如果由于某些原因,当前项的URL不能获取到,它将使songLabel显示字符串"Unknown".

你也需要一个去更新timeLabel的方法来显示当前播放项消耗的时间.使用addPeriodicTimeObserverForInterval(\_:queue:usingBlock:)是达到当前目的最好的方法,该函数讲调用给定的队列当中提供的块.在viewDidLoad()的末尾处添加如下代码:

```

player.addPeriodicTimeObserverForInterval(CMTimeMake(1, 100), queue: dispatch_get_main_queue()) {
    [unowned self] time in
    let timeString = String(format: "%02.2f", CMTimeGetSeconds(time))
}
if UIApplication.sharedApplication().applicationState == .Active {
    self.timeLabel.text = timeString
} else {
    println("Background: \(timeString)")
}
}

```

这添加给播放器一个周期性的观察者,如果这个APP在前台,这个观察者每一秒的1/100就会被调用一次并且更新UI.

**重要提示:**由于你想在结束时更新UI,你必须确保这些代码在主队列中被调用.这就是你指定dispatch\_get\_main\_queue()参数的原因.

在这里暂停一下,思考应用的状态.

你的应用处于下面五个状态之一中.简单地说,他们是:

\*未运行:你的APP在开启之前处于这个状态.

\*激活:一旦你的APP被开启,它变成活跃状态.

\*未激活:当你的APP正在运行,但是一些事情打断它的动作,比如有电话打进来,它变成inactive状态.休眠意味着这个APP仍然在前台运行,只是它没有接收事件.

\*后台:在这个状态下,你的APP不在前台显示了但是它仍然在执行代码.

\*挂起:你的APP进入不再运行代码的状态.

如果你想更深入的了解这些状态之间的区别,苹果网站的Execution States for Apps对此有很详细介绍.

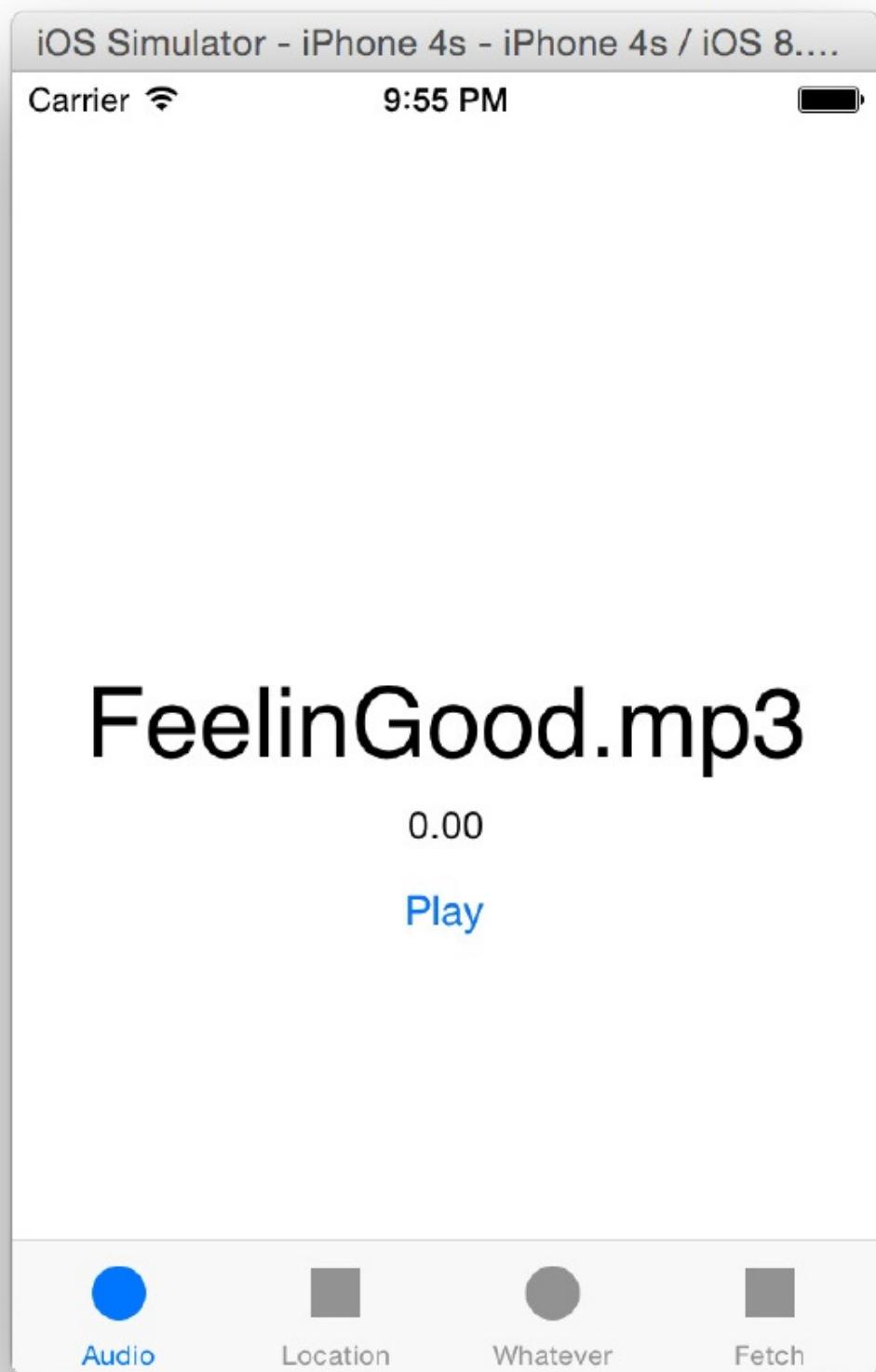
你可以通过读取UIApplication.sharedApplication().applicationState来检查APP的转台.记住:你只能获取三种状态的返回值: .Active, .Inactive, and .Background.当你的APP在执行代码的时候,挂起状态和未运行状态很明显不可能出现.

让我们将目光继续放在之前代码上,如果该应用处于激活状态,你需要更新音乐标题栏.在后台中,你仍然能够更新这个label的文字,但是这点知识证明了当你APP在后台的时候继续接受回调.

现在,把剩余的代码添加到`playPauseAction(:)`的实现中,让播放/暂停按钮工作.在`AudioViewController`中,把下面代码添加到`playPauseAction(:)`的实现中:

```
@IBAction func playPauseAction(sender: UIButton) {
    sender.selected = !sender.selected
    if sender.selected {
        player.play()
    } else {
        player.pause()
    }
}
```

很好,这是你全部的代码.创建并运行,你将看到下面的样子:



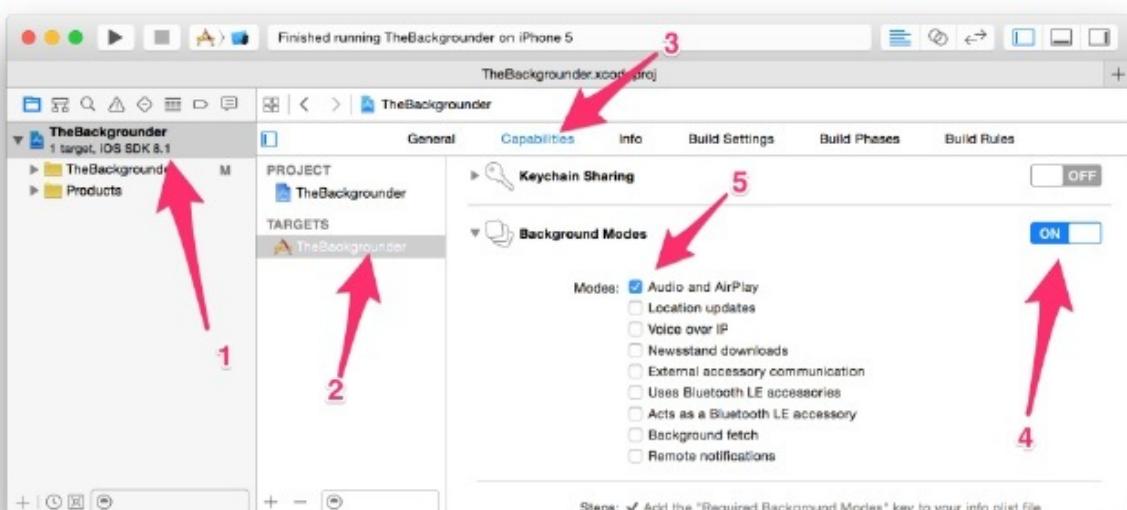
现在,点播放,音乐将开始.很好!

测试后台模式是否起作用.按home按钮(如果你正在使用模拟器,按Cmd-Shift-H).如果你在真正的设备上运行(不是Xcode 6.1的模拟器)音乐将停止.这是为什么呢?还有很重要的一块落下了!)

对于大多数的后台模式("Whatever"模式除外)你需要在Info.plist中添加一个key用来指明APP在后台中运行的代码.幸运的是,在Xcode6可以通过复选框进行选择.

回到Xcode,按照以下步骤进行操作:

1. 在项目管理器中点击工程
2. 点击目标TheBackgrounder
3. 点击功能标签
4. 滑动背景模式并设置为ON
5. 选中 Audio和AirPlay



重新编译并且运行.开始运行音乐并且点击home键,尽管这个APP在后台运行,这次你就会依旧能够听到音乐.

You should also see the time updates in your Console output in Xcode, proof that your code is still working even though the app is in the background. You can download the partially finished sample project up to this point.

在Xcode的输出里你也能够在控制台看到实时的更新,着就证明了虽然你的APP在后台运行,但是你的代码依旧在工作.现在你可以[下载部分完成的示例代码了](#).

以上第一个模式结束了,如果你想学完整个教程--那就继续往下读吧!

## 接收位置更新

当在后台模式进行定位时，你的APP依旧会随着用户更新位置而接收到位置信息，甚至APP在后台的时候.你可以控制这些位置更新的准确性,甚至改变精度.

如果你的app真正需要这些信息来为用户提供价值，你只能使用后台模式.如果你使用这个模式并且Apple看到用户将要获得这些信息，你的应用程序将会被拒绝.有时苹果也将要求你向app添加一个警告的描述说明app将导致增加电量的使用.

第二步是为了位置更新，打开LocationViewController.swift并且向里面增加一些属性用来初始化LocationViewController.

```
var locations = [MKPointAnnotation]()

lazy var locationManager: CLLocationManager! = {
    let manager = CLLocationManager()
    manager.desiredAccuracy = kCLLocationAccuracyBest
    manager.delegate = self
    manager.requestAlwaysAuthorization()
    return manager
}()
```

你将使用locations来存储能够绘制在地图上的位置信息.CLLocationManager可以使你能够从设备上获取位置更新.你使用延迟的方法实例化它,所以当你第一次访问该属性被调用的函数时,它才被初始化.

代码可以设置位置管理器的精确度来实现最高的精确，你可以调节到你的app所需要的精确度.你会了解更多关于其他精度设置和它们的重要性.注意你也可以调用requestAlwaysAuthorization().这是在IOS8中的要求，并且为用户提供了接口来允许用户在后台使用位置.

现在你可以填写空的accuracyChanged(\_:)的实现在LocationViewController里:

```

@IBAction func accuracyChanged(sender: UISegmentedControl) {
    let accuracyValues = [
        kCLLocationAccuracyBestForNavigation,
        kCLLocationAccuracyBest,
        kCLLocationAccuracyNearestTenMeters,
        kCLLocationAccuracyHundredMeters,
        kCLLocationAccuracyKilometer,
        kCLLocationAccuracyThreeKilometers]

    locationManager.desiredAccuracy = accuracyValues[sender.selectedSegmentIndex]
}

```

accuracyValues是由CLLocationManager的desiredAccuracy可能值构成的数组.这些变量控制了你的位置的精确度.

你可能认为这种方式是愚蠢的.为什么位置管理器不能够给你最精确的位置信息呢?最重要的原因是是为了节省电量.低精确意味着耗电量较低.

这就意味着你应该选择最少的值实现你的app可以承受的最低限度的精确度.你随时可以修改这些值在你的需求.

另一个性能就是你可以控制你的app接收位置更新的频率,忽视desiredAccuracy:distanceFilter的值.当你的设备移动到了一定的值(以米计算)时,这个性能告诉位置管理器你只想接收位置更新.这个值应该最大限度的节省你的电池消耗.

现在你可以在enabledChanged(:)中添加代码来实现获取位置更新:

```

@IBAction func enabledChanged(sender: UISwitch) {
    if sender.on {
        locationManager.startUpdatingLocation()
    } else {
        locationManager.stopUpdatingLocation()
    }
}

```

这个代码示例有一个与动作相关的UISwitch,这个UISwitch实现了位置跟踪的开启与关闭.

下一步你可以通过添加一个`CLLocationManagerDelegate`方法来接收位置更新.添加以下方法到`LocationViewController`中.

```
// MARK: - CLLocationManagerDelegate

func locationManager(manager: CLLocationManager!, didUpdateToLocation newLocation: CLLocation!, fromLocation oldLocation: CLLocation!) {
    // Add another annotation to the map.
    let annotation = MKPointAnnotation()
    annotation.coordinate = newLocation.coordinate

    // Also add to our map so we can remove old values later
    locations.append(annotation)

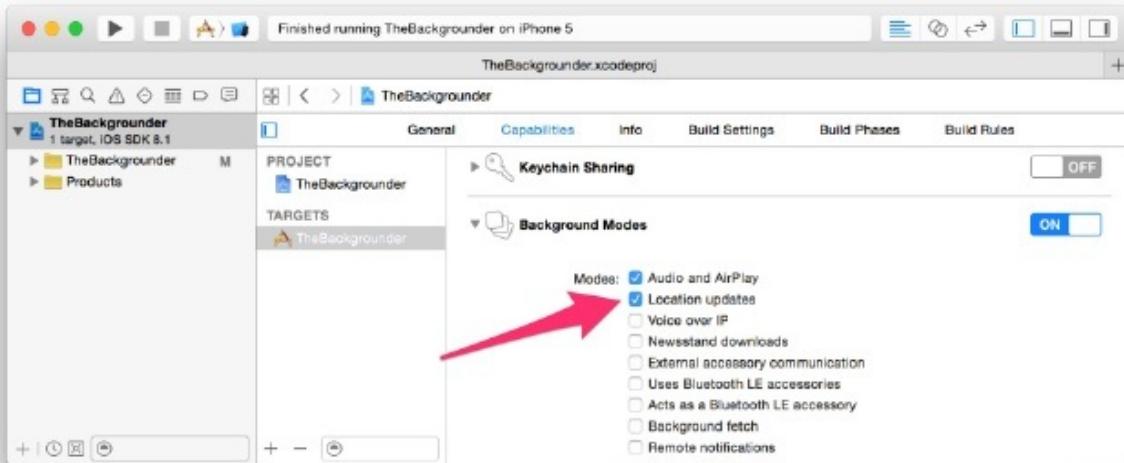
    // Remove values if the array is too big
    while locations.count > 100 {
        let annotationToRemove = locations.first!
        locations.removeAtIndex(0)

        // Also remove from the map
        mapView.removeAnnotation(annotationToRemove)
    }

    if UIApplication.sharedApplication().applicationState == .Active {
        mapView.showAnnotations(locations, animated: true)
    } else {
        NSLog("App is backgrounded. New location is %@", newLocation)
    }
}
```

如果app的状态是激活状态，这些代码将更新地图.如果这个app在后台运行，你应该在xcode的控制台来看位置更新的log.

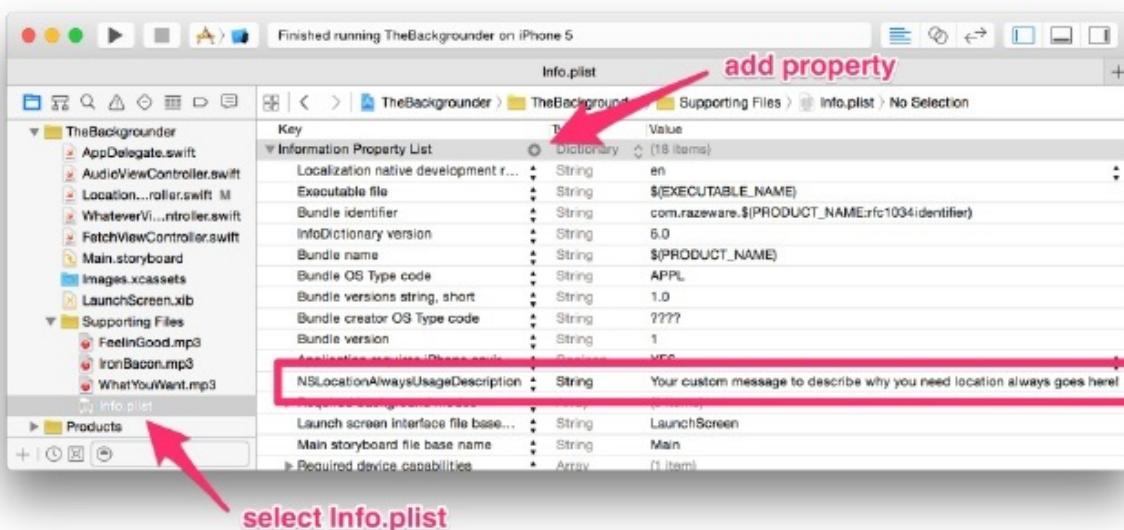
现在你已经知道了后台模式，现在你不应该犯以前的相同的错误了.现在你可以在`Location updates`中设置使得ios知道你的app想在后台运行时继续接受位置更新.



除了更改这个之外，你应该在你的Info.plist中设置一个关键词来允许你向使用者解释为什么后台更新数据是需要的。如果不被允许后台更新，位置更新就会慢慢地失败。

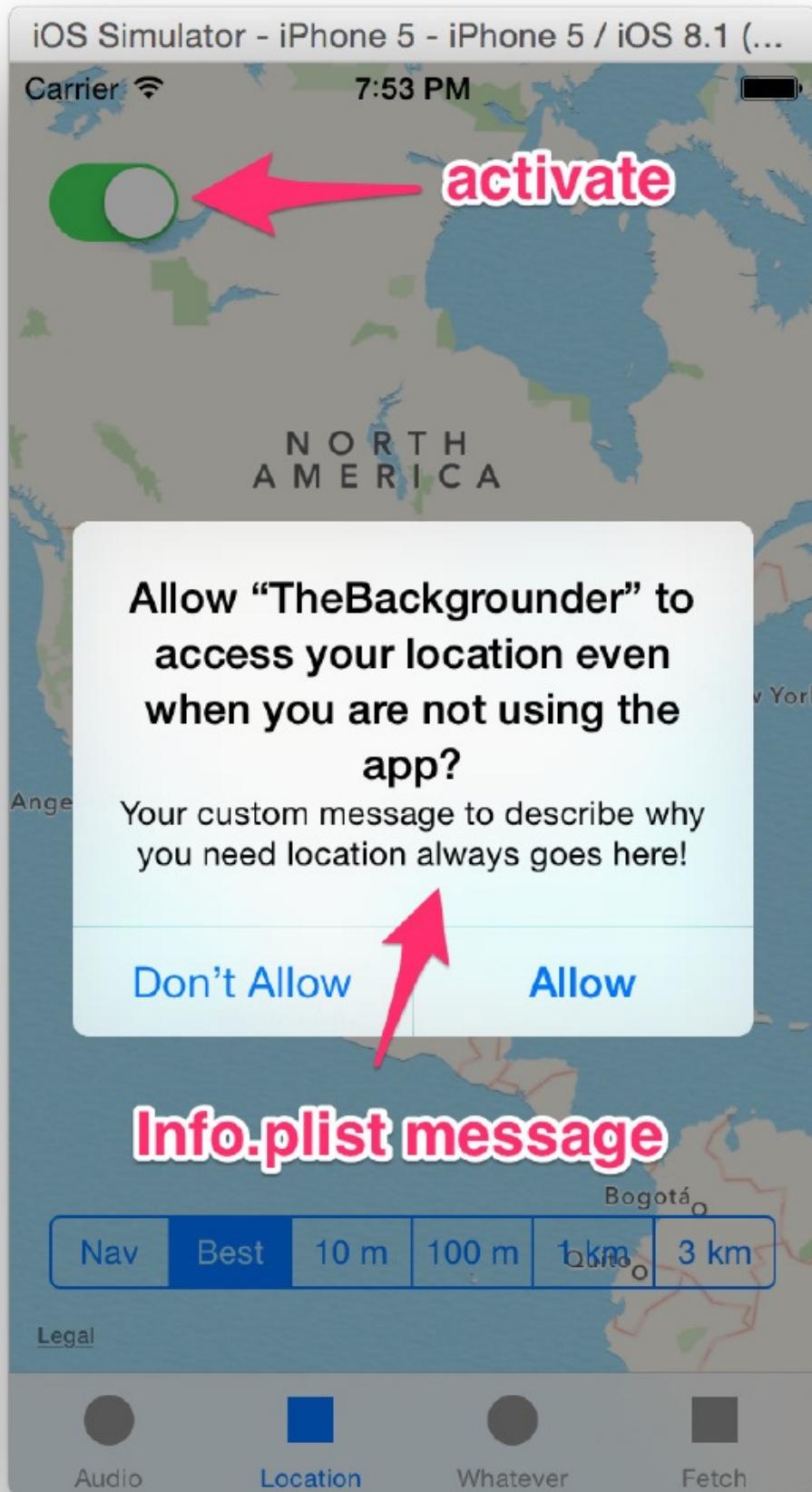
步骤如下：

1. 选择Info.plist文件
2. 点击+号来添加一个关键词
3. 点击这个关键词的名字：NSLocationAlwaysUsageDescription
4. 描述为什么你需要在后台位置更新，能够另使用者信服。

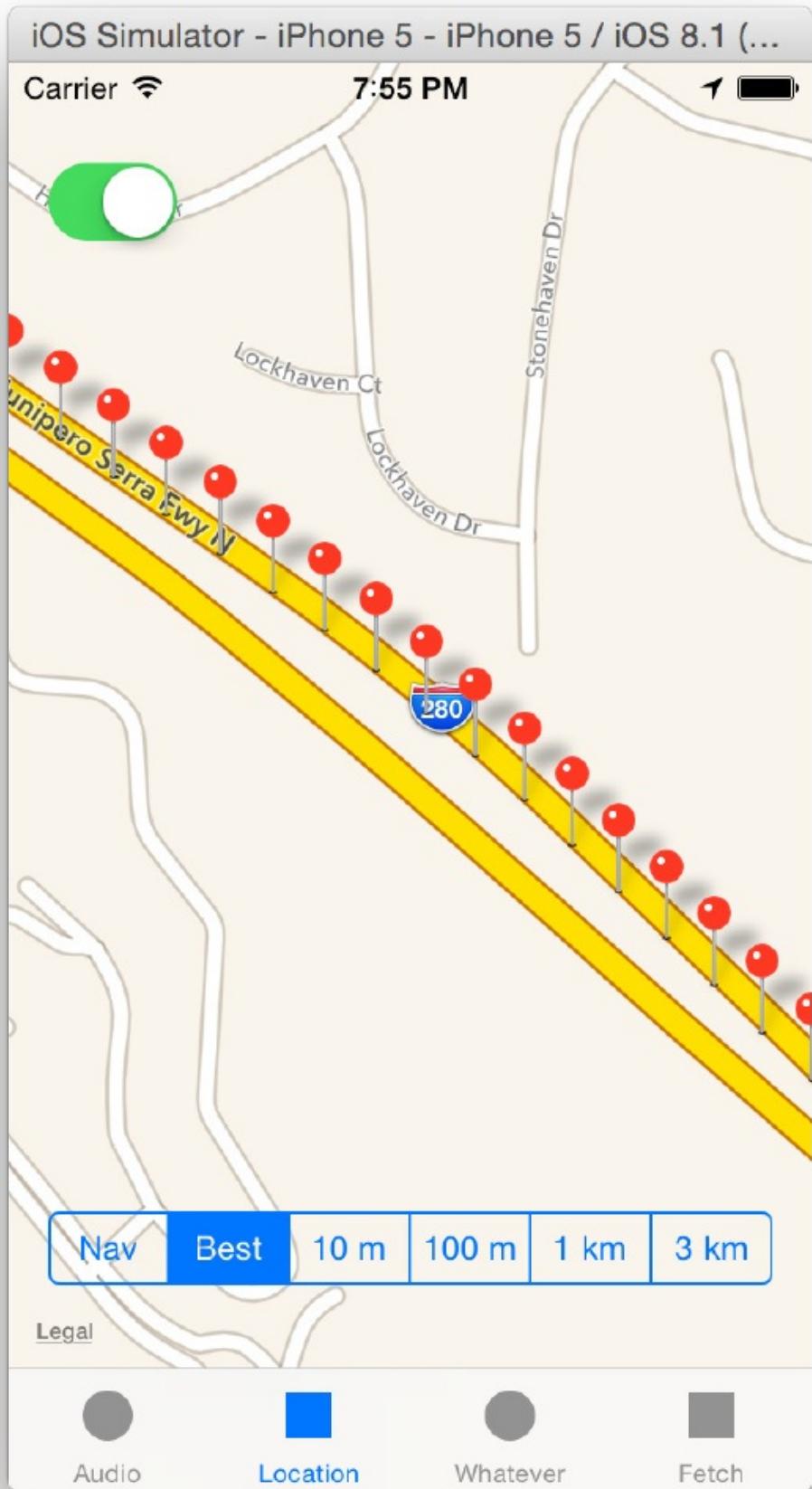


现在你可以编译并且运行你的程序了。切换到第二个选项卡并打开开关。

当你第一次运行的时候，你会看到你写入到Info.plist中的信息.点击allow出去走走，或者围绕你周围的建筑转一转.这时候你就开始看到位置信息的更新，在模拟器里也可以实现.

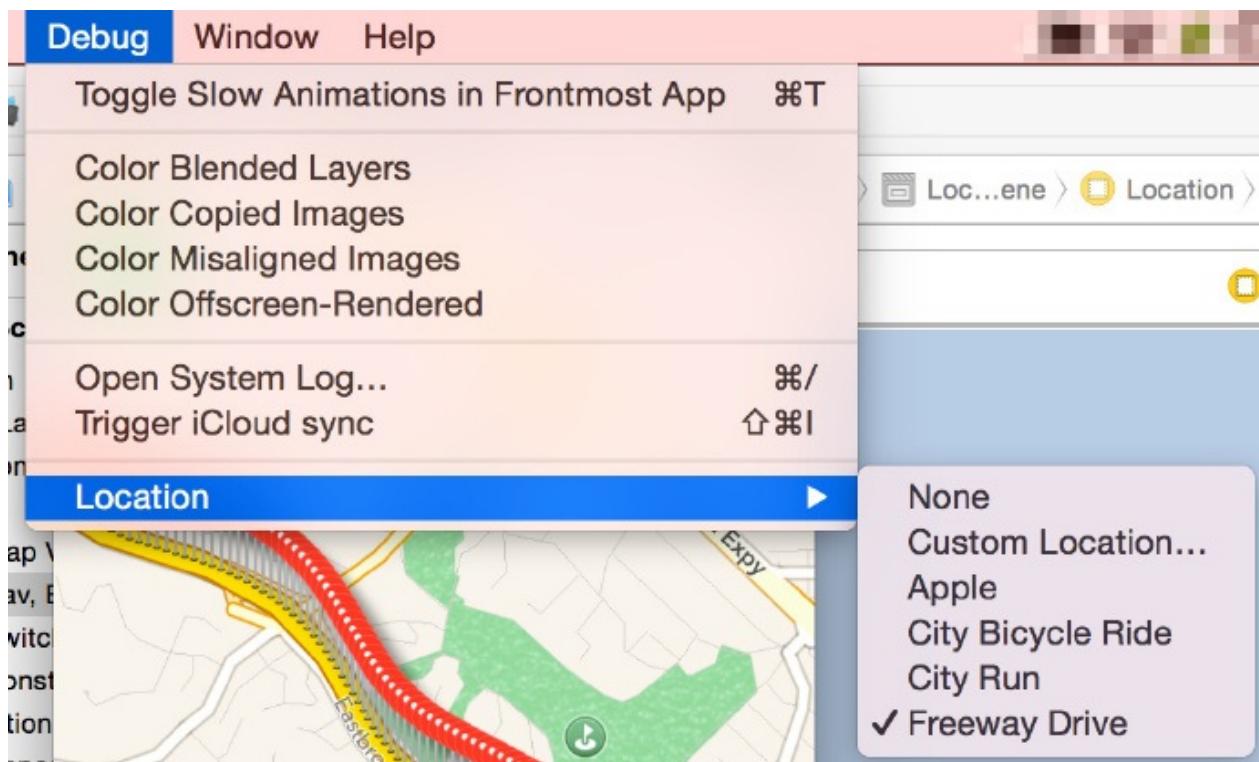


过一会，你将会看到如下的一些东西：



如果你在后台运行你的app，你将会在你的控制台log看到你的app位置更新信息。重新打开你的app，你就会发现地图上有所有的位置点，这些就是你的app在后台运行时候更新的数据。

如果你使用的是模拟器，你也可以使用这个app来模拟这个动作。打开菜单Debug \ Location：



设置location选项为Freeway Drive然后点击home按钮。这时候你就会看到在控制台打印出你的程序运行的状态，就像你在模拟你开车在加利福尼亚的高速公路上。

```
2014-12-21 20:05:13.334 TheBackgrounder[21591:674586] App is bac  
kgrounded. New location is <+37.33482105, -122.03350886> +/- 5.00  
m (speed 15.90 mps / course 255.94) @ 12/21/14, 8:05:13 PM Pacific Standard Time  
2014-12-21 20:05:14.813 TheBackgrounder[21591:674586] App is bac  
kgrounded. New location is <+37.33477977, -122.03369603> +/- 5.00  
m (speed 17.21 mps / course 255.59) @ 12/21/14, 8:05:14 PM Pacific Standard Time  
2014-12-21 20:05:15.320 TheBackgrounder[21591:674586] App is bac  
kgrounded. New location is <+37.33474691, -122.03389325> +/- 5.00  
m (speed 18.27 mps / course 257.34) @ 12/21/14, 8:05:15 PM Pacific Standard Time  
2014-12-21 20:05:16.330 TheBackgrounder[21591:674586] App is bac  
kgrounded. New location is <+37.33470894, -122.03411085> +/- 5.00  
m (speed 19.27 mps / course 257.70) @ 12/21/14, 8:05:16 PM Pacific Standard Time
```

现在你可以下载这个示例程序了,到第三个选项卡和第三个后台模式.

## 执行有限长任务等

下一个后台模式在可以正式的称为后台执行有限长的任务 (Executing a Finite-Length Task in the Background) .

严格的说这并不是真正意义上的后台模式，因为你并没有在Info.plist中声明在你的app中使用这个模式（或者在复选框中使用Background Mode）.相反，它只是一个api你可以让你的任意代码运行有限的时间，当你的app在后台运行的时候.

在过去，这个模式只是在上传或者下载或者运行某一段时间来完成某一项任务.但是如果这个链接很缓慢或者这个进行一直不结束怎么办？它会让你的应用程序在一个奇怪的状态,你必须添加大量的代码来处理错误使得程序稳健地工作.因为这样的原因，Apple介绍了NSURLSession.

NSURLSession在面对后台运行甚至设备重启时具有鲁棒性，并且以减少设备能耗的方式完成任务.如果你想处理大规模的下载，请查看我们的[NSURLSession tutorial](#).

这种后台运行模式对完成一些长时间的任务还是一种非常有效的方法，比如在相机相册中进行渲染和写入一个视频。



但是这只是一个例子。你可以运行的代码是任意的，你可以用这个api来实现任意的事情：运行长时间的计算，将过滤器应用到图像处理，渲染一个复杂3d网格...whatever！只要是你想在长时间运行你的程序你都可以用这个api。

你的app在后台运行的时间取决于ios系统。对于后台运行时间你可以在UIApplication中查询backgroundTimeRemaining，它将会告诉你剩余多长时间。

一般来说你会有3分钟时间来实现。但是在api文档中并没有给一个大约的时间，所以你不能依赖这个时间，可能是5分钟也可能是5秒。所以你的app需要准备发生的任何事情。

这里给一个计算机学生都熟悉的任务：斐波纳契数列。

这里的含义是，你会在后台计算这些数字！

打开WhateverViewController.swift并且在WhateverViewController里面添加属性。

```
var previous = NSDecimalNumber.one()
var current = NSDecimalNumber.one()
var position: UInt = 1
var updateTimer: NSTimer?
var backgroundTask: UIBackgroundTaskIdentifier = UIBackgroundTaskIdentifier(rawValue: "com.yourcompany.yourapp.backgroundtask")
```

NSDecimalNumbers将保存序列中的前两个数的值。NSDecimalNumbers可以保存大的数据，因此非常适合你的目标。Position只是一个计数器来告诉你这个数在当前序列中的位置。

你将使用updateTimer证明甚至计时器继续使用这个API时,也稍微放慢速度的计算,这样你就可以观察他们.

在WhateverViewController中添加一些实用方法来重置斐波那契计算,启动和停止能够后台运行的任务:

```
func resetCalculation() {
    previous = NSDecimalNumber.one()
    current = NSDecimalNumber.one()
    position = 1
}

func registerBackgroundTask() {
    backgroundTask = UIApplication.sharedApplication().beginBackgroundTaskWithExpirationHandler {
        [unowned self] in
        self.endBackgroundTask()
    }
    assert(backgroundTask != UIBackgroundTaskInvalid)
}

func endBackgroundTask() {
    NSLog("Background task ended.")
    UIApplication.sharedApplication().endBackgroundTask(backgroundTask)
    backgroundTask = UIBackgroundTaskInvalid
}
```

现在到了重要部分，在didTapPlayPause(\_:)添加空的实现：

```

@IBAction func didTapPlayPause(sender: UIButton) {
    sender.selected = !sender.selected
    if sender.selected {
        resetCalculation()
        updateTimer = NSTimer.scheduledTimerWithTimeInterval(0.5, target
            : self,
            selector: "calculateNextNumber", userInfo: nil, repeats: true)
        registerBackgroundTask()
    } else {
        updateTimer?.invalidate()
        updateTimer = nil
        if backgroundTask != UIBackgroundTaskInvalid {
            endBackgroundTask()
        }
    }
}

```

按钮改变选择状态取决于计算已经停止，应该开始或者是计算已经开始，应该停止。

首先你必须设置斐波那契序列变量。然后你可以创建一个NSTimer，没秒启动两次，并且调用 calculateNextNumber() 函数。

现在到了一个重要的时刻：调用 registerBackgroundTask() 函数，反过来调用 beginBackgroundTaskWithExpirationHandler(\_:)。这个方法告诉了 iOS 你需要时间在后台运行你的 app。这些调用完成之后，在你调用 endBackgroundTask() 之前你的 app 会一直获取 CPU 时间。

嗯，差不多。如果你的 app 在后台运行一段时间后没有调用 endBackgroundTask()，iOS 将调用关闭程序定义，这是在你调用 beginBackgroundTaskWithExpirationHandler(\_:) 时给你机会来停止执行代码。所以调用 endBackgroundTask() 告诉 iOS 你已经完成工作了是非常好的一个主意。如果你不执行上面所说的而是继续执行你的代码，你的 app 将会终止。

第二部分关于 if 的语句是很简单的：它只是使定时器失效，并且调用 endBackgroundTask() 来告诉 iOS 不再需要额外的 CPU 时间。

在你每次调用 beginBackgroundTaskWithExpirationHandler(\_) 时调用 endBackgroundTask() 是非常重要的。如果你在一个任务里调用 beginBackgroundTaskWithExpirationHandler(\_) 两次而只调用

endBackgroundTask()一次，你将仍然获取cpu时间，直到你在运行第二次的后台任务是调用endBackgroundTask()才能结束.这就是为什么你需要backgroundTask.

现在你可以实现简单的计算机程序方法.在WhateverViewController添加以下的方法：

```
func calculateNextNumber() {
    let result = current.decimalNumberByAdding(previous)

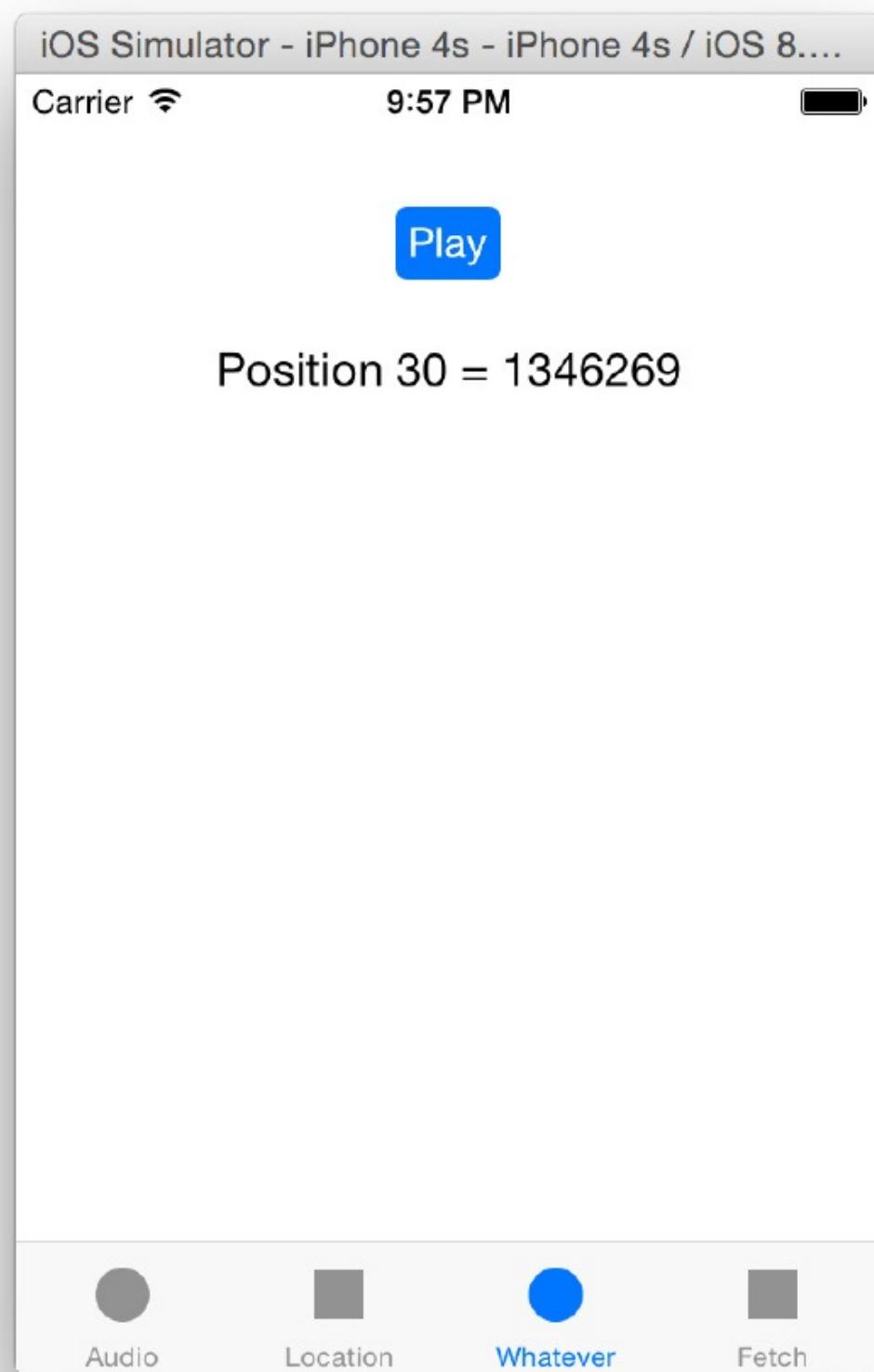
    let bigNumber = NSDecimalNumber(mantissa: 1, exponent: 40, isNegative: false)
    if result.compare(bigNumber) == .OrderedAscending {
        previous = current
        current = result
        ++position
    }
    else {
        // This is just too much.... Start over.
        resetCalculation()
    }

    let resultsMessage = "Position \(position) = \(current)"

    switch UIApplication.sharedApplication().applicationState {
    case .Active:
        resultsLabel.text = resultsMessage
    case .Background:
        NSLog("App is backgrounded. Next number = %@", resultsMessage)
        NSLog("Background time remaining = %.1f seconds", UIApplication.
sharedApplication().backgroundTimeRemaining)
    case .Inactive:
        break
    }
}
```

再一次，我们将展示另一个方法即使你的app在后台运行依旧能够显示结果.在这种情形下，还有一个有趣的信息: backgroundTimeRemaining的数值.只有当ios调用添加调用beginBackgroundTaskWithExpirationHandler(\_:)时才会停止.

编译并且运行，然后切换到第三个选项卡。



点击play并且你将会看到app计算出的值.现在点击home键然后查看xcode控制台.你应该会看到app依旧会更新数字，与此同时时间依旧在向前走.

在大多数情况下，这个时间将从第180秒开始并且延续5秒钟.如果你等待重新回到你的app，定时器将重新开始启动并且所有的错误行为将继续.

在代码里只有一个bug，它给我机会来解释关于后台通知.假设你或太运行app并且等待分配的时间到期.在这种情况下，你app将调用？？并且调用*endBackgroundTask()*，也就是终结后台运行时间的需求.

如果你继续返回你的app，定时器将继续激活.但是如果你离开app，你将不会得到或太运行时间.Why？因为在超时和回到后台期间app没有间隙来调用*beginBackgroundTaskWithExpirationHandler(\_)*.

你怎么解决这个问题呢？有许多方法能够解决这个问题，并且其中一个是使用一种状态来改变通知.

有两种你可以得到通知并且你的app可以改变它的状态的方法：第一种是通过你的主app委托方法；第二种是通过监听ios发送给你的app的通知.

\* 当你的app将要进入不活跃的状态，*UIApplicationWillResignActiveNotification*和*applicationWillResignActive(\_)*将会被发送和调用.在这种情况下，你的app不是在后台运行，它依旧在前台运行，但是它将不会接收到任何UI事件.

\* 当app进入到后台状态，*UIApplicationDidEnterBackgroundNotification* 和*applicationDidEnterBackground(\_)*将会被发送和调用.在这种情况下，你的app将不会是在激活状态，并且它是你最后的机会运行你的代码.如果你想得到更多的CPU时刻，这是一个调用*beginBackgroundTaskWithExpirationHandler(\_)*非常完美的时机.

\* 当app返回激活状态，*UIApplicationDidEnterForegroundNotification* 和*applicationDidEnterForeground(\_)*将会被发送和调用.这是app依旧在后台运行，你已经可以启动任何你想做的事.当你真正进入后台运行是如果你只调用了*beginBackgroundTaskWithExpirationHandler(\_)*，此时将是一个好的时机调用*endBackgroundTask()*.

\* 以防你的app从后台运行状态返回，在前一个通知完成后*UIApplicationDidBecomeActiveNotification*和*applicationDidBecomeActive(\_)*将会被发送和调用.如果你的app只是临时的中断也会被调用-举例—如果你的app没有真正的进入到后台，但是你依旧会收到*UIApplicationWillResignActiveNotification*.

你可以在[Apple's documentation for App States for Apps](#)中看到所有的图像化描述  
(文章—有着许多非常棒的图表)

现在是解决这个bug的时间了.首先要重写viewDidLoad()并且订阅  
UIApplicationDidBecomeActiveNotification.

```
override func viewDidLoad() {
    super.viewDidLoad()
    NSNotificationCenter.defaultCenter().addObserver(self, selector:
        Selector("reinstateBackgroundTask"), name: UIApplicationDidBecomeActiveNotification, object: nil)
}
```

不管何时这app变成激活状态，指定的选择器reinstateBackgroundTask将被调用.

不管何时你订阅了一个通知你也应该想到这个订阅的通知哪里不应该被订阅.使用  
deinit来完成这个功能.按照下面的代码加入到WhateverViewController.

```
deinit {
    NSNotificationCenter.defaultCenter().removeObserver(self)
}
```

最后实现reinstateBackgroundTask().

```
func reinstateBackgroundTask() {
    if updateTimer != nil && (backgroundTask == UIBackgroundTaskInvalid) {
        registerBackgroundTask()
    }
}
```

如果定时器依然运行但是后台任务没有运行，你只需要恢复就可以了.

把你的代码分解成小的实用的代码只需要做一件事就可以.当一个后台任务不是在当前的定时器下你只需要调用registerBackgroundTask()即可.

然后你可以使用了.你可以[下载](#)这个程序.

这个课程的最后一节是 : Background Fetching.

## 后台获取

后台获取是iOS7中推出的让你的APP在最大限度减少对电池损耗的时候总是展现最新的信息.举个例子,假设你正在给你的APP填充信息.你可以通过 `viewWillAppear(_:)`. 获取最新数据来预先通知后台模式.这个方案可以解决在新数据刷新过来之前你的用户正在浏览前几秒的数据.当用户打开你APP的同时,最新的数据同时被神奇的展现了,这种情况再好不过了.这是后台模式能够为你实现的操作.

当APP被激活的时候,系统会使用惯用模式去决定什么时候执行后台获取.比如,如果用户每天都在早上9点打开改APP,后台获取在这个时间点之前预先执行是很可能的.系统决定什么时候是安排后台获取的最好时间,因此你不应该用它去做紧急的更新.

这里有你为了实现后台获取必须做的三件事情:

- \* 检查你APP Capabilities 选项中后台模式的后台获取选项框是否被选中.
- \* 使用 `setMinimumBackgroundFetchInterval(_:)` 为你的APP创建一个合适的时间间隔.
- \* 在你APP委托中实现 `application(_:performFetchWithCompletionHandler:)` 去管理后台获取.

后台获取就像他名字表示的一样,他通常涉及到从外源,比如网络服务,中获取信息.就这个教程的意图,你将不会使用网络而仅仅获取现在的时间.这样简化讲让你理解在不同担心外在的服务的时候操作并测试后台模式所需要的每一样东西.

对于有限长度的任务,你只有以按秒为单位的时间去执行操作,公认的时间是不超过30秒,但越短越好.如果您需要下载大量资源最为获取的部分,这就是你需要使用NSURLSession的背景传输服务的地方.

开始的时间到了.首先,打开 `FetchViewController.swift`,并将下面的属性和方法添加到 `FetchViewController` 中.

```

var time: NSDate?

func fetch(completion: () -> Void) {
    time = NSDate()
    completion()
}

```

这些代码是代替你真正的从外源(json或XML RESTful 服务)中获取数据的一种简化.因为它可能需要几秒钟来获取和分析数据,你传递一个完成的handler,这个handler在进程完成后被调用.你待会儿会看到为什么很很重要.

接下来,完成view controller的代码.将下面的方法添加到FetchViewController中.

```

func updateUI() {
    if let time = time {
        let formatter = NSDateFormatter()
        formatter.dateStyle = .ShortStyle
        formatter.timeStyle = .LongStyle
        updateLabel?.text = formatter.stringFromDate(time)
    }
    else {
        updateLabel?.text = "Not yet updated"
    }
}

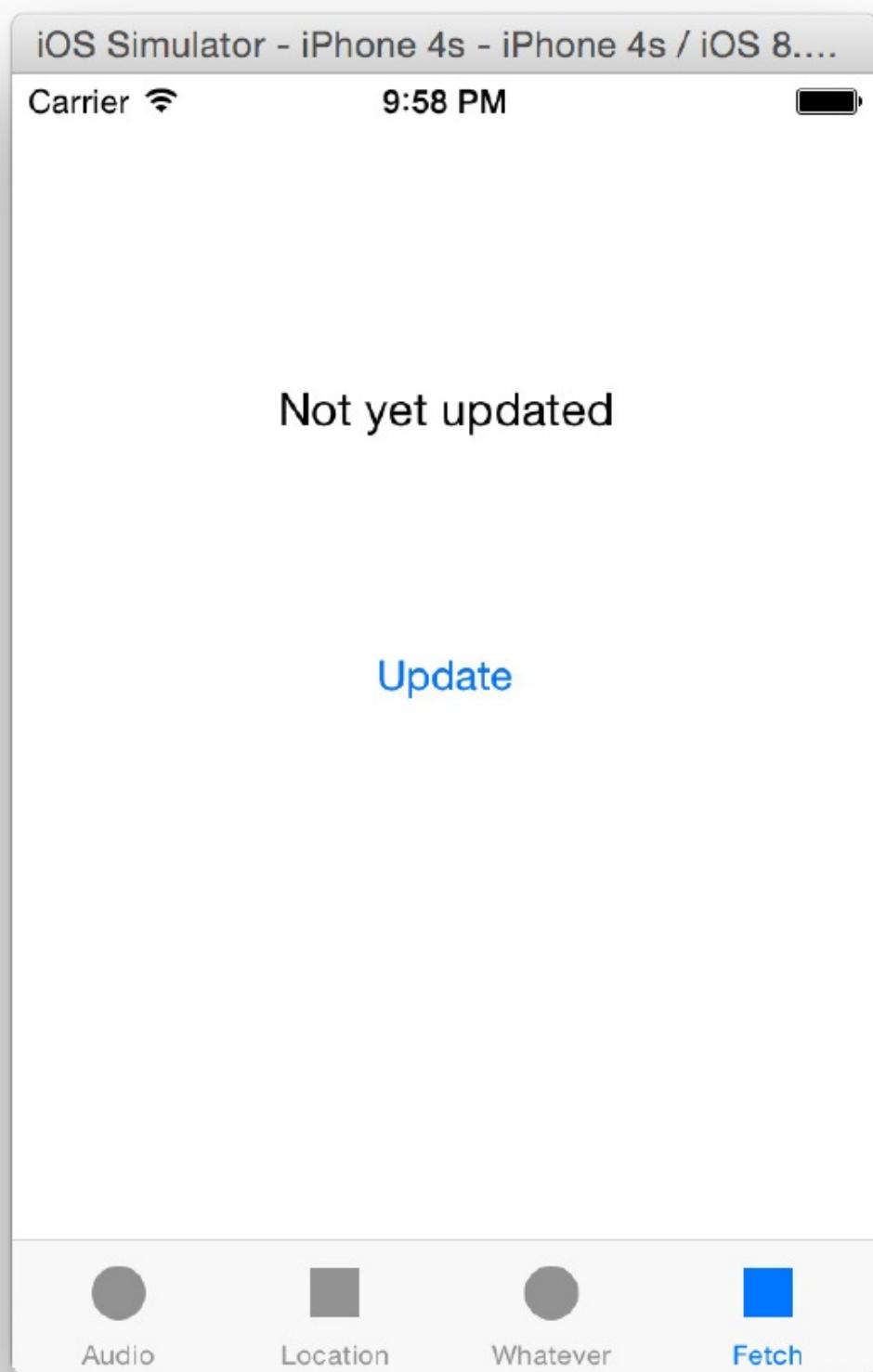
override func viewDidLoad() {
    super.viewDidLoad()
    updateUI()
}

@IBAction func didTapUpdate(sender: UIButton) {
    fetch { self.updateUI() }
}

```

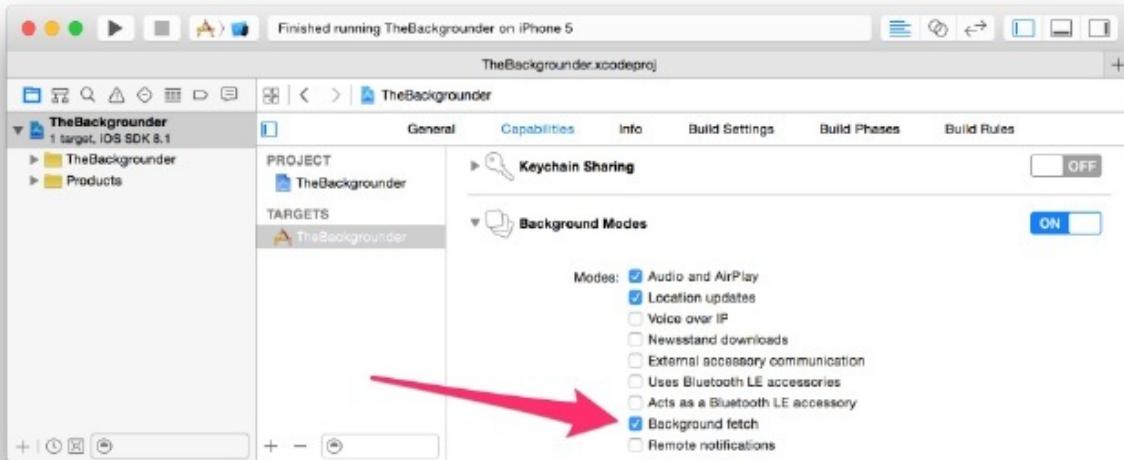
updateUI()格式化这个时间并显示它.它是一个可选的类型,所以如果它没有被创建,他将展示至今没有更新的信息.当这个view初次被加载时(在 viewDidLoad()中)你不能获取到,但是直接调用updateUI()函数,将会有"Not yet updated"的字样在开始时显示.最后,当更新按钮被监听的时候,它运行获取的代码并且会完成对UI的更新.

就这一点而言,该view controller正在工作.



然而,后台获取没有起作用.

启用后台获取的第一步是在**Capabilities**选项栏里选中**Background fetch**.到现在这个操作已经是老一套的了,直接找到它并选中.



接下来,打开**AppDelegate.swift**,通过在  
`application(_:didFinishLaunchingWithOptions:)`中设置最小的后台获取时间间隔来  
请求后台获取操作.

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    UIApplication.sharedApplication().setMinimumBackgroundFetchInterval(
        UIApplicationBackgroundFetchIntervalMinimum)

    return true
}
```

默认的时间间隔是你想切换回去的**UIApplicationBackgroundFetchIntervalNever**,比如,你的用户日志和不需要更新的内容.你也可以设置一个精确到秒的时间间隔.系统在开始执行后台获取之前将等待一段时间.

要小心,,不要将时间间隔设置过短,因为它会多余的消耗电池和损害服务器.结束获取信息的确切时间是由系统决定的,但是在执行它之前将会等待一段时间.通常,**UIApplicationBackgroundFetchIntervalMinimum**是很好用的默认值.

最后,为了启用后台程序,你必须实现

`application(_:performFetchWithCompletionHandler:)`.将下列方法添加到 `AppDelegate.swift` 中.

```
// Support for background fetch
func application(application: UIApplication, performFetchWithCompletionHandler completionHandler: (UIBackgroundFetchResult) -> Void) {

    if let tabBarController = window?.rootViewController as? UITabBarController,
        viewControllers = tabBarController.viewControllers as? [UIViewController] {
        for viewController in viewControllers {
            if let fetchViewController = viewController as? FetchViewController {
                fetchViewController.fetch {
                    fetchViewController.updateUI()
                    completionHandler(.NewData)
                }
            }
        }
    }
}
```

首先你需要获取 `FetchViewController`.然后,因为 `rootViewController` 在每个APP中不是必须的 `UITabBarController`,所以它是可以选择创建的,不过它在这个APP中,所以它绝不会出现问题.

接下来,你在选项卡控制器中循环添加所有的视图控制器,并且将它们成功的放到 `FetchViewController` 中.在这个APP中,你知道它是最后的控制器,所以你不能对它进行硬编码,但是在你决定以后添加或删除选项卡的时候循环创建会提高程序的健壮性.

最后,你可以调用 `fetch(_:)`.当它执行完后,你会更新UI,然后调用将 `completionHandler` 作为参数传递的函数.你在这个操作的最后调用这个完成处理的程序是很重要的.你指定在获取过程中获取的结果作为以一个参数.它的可能值为 `.NewData`, `.NoData` 或者 `.Failed`.

为了简单起见,该教程总是指定.NewData作为永远成功获取时间的返回值,并且这个值和上一次的结果总是不同的.在这之后,iOS可以使用更好的时间间隔来执行后台获取.该系统知道在这个时间点上的系统快照,所以它可以在应用程序切换卡中显示.以上是为了实现后台获取所需要的所有的操作.

提示:不是沿着信息传递完成对属性的调用,而是保存一个属性变量,并且在你获取完成后调用它是很有诱惑力的.不这样做的话,如果你多次调用application(\_:performFetchWithCompletionHandler:),先前的处理程序将会被覆盖,永远不会被调用.最好通过传递处理器,并且在它不会造成这种编程错误的时候调用它.

## 测试后台获取

测试后台获取的一个方法是停下来等着系统决定去执行它.这需要大量的等待.幸运的是,Xcode体统了模拟后台获取的方法.有两种你需要测试的情况,一种是当你的APP在后台中时,另一种是你的APP处于从被挂起到继续运行的情况.第一种方法最简单,仅仅是一个选择菜单.

- \* 在真正的设备上运行(不是模拟器);
- \* 在Xcode调试菜单中选择模拟后台获取;

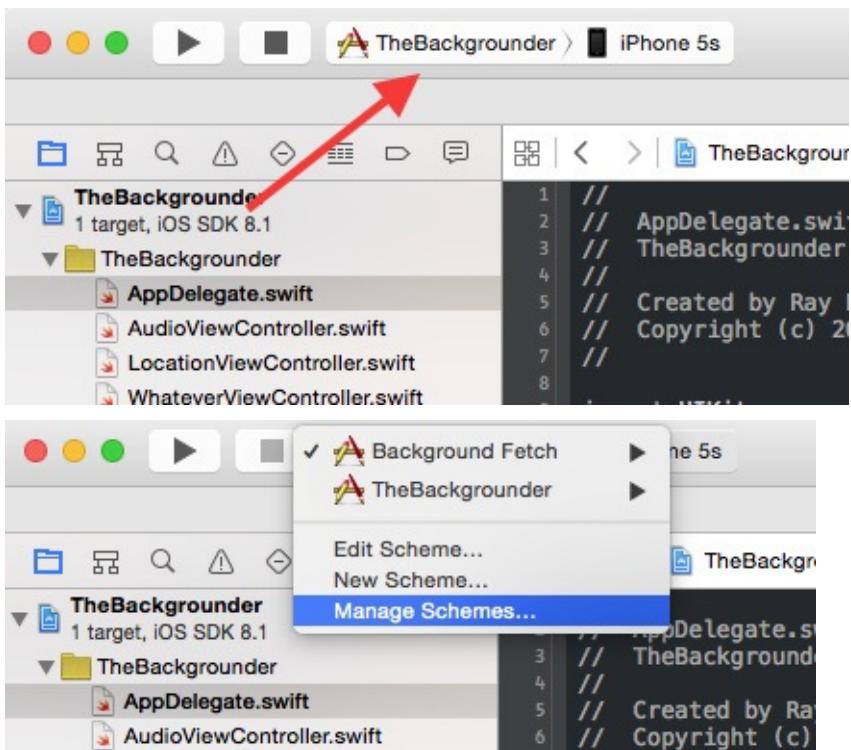


重新打开这个APP,注意被送到后台的数据.

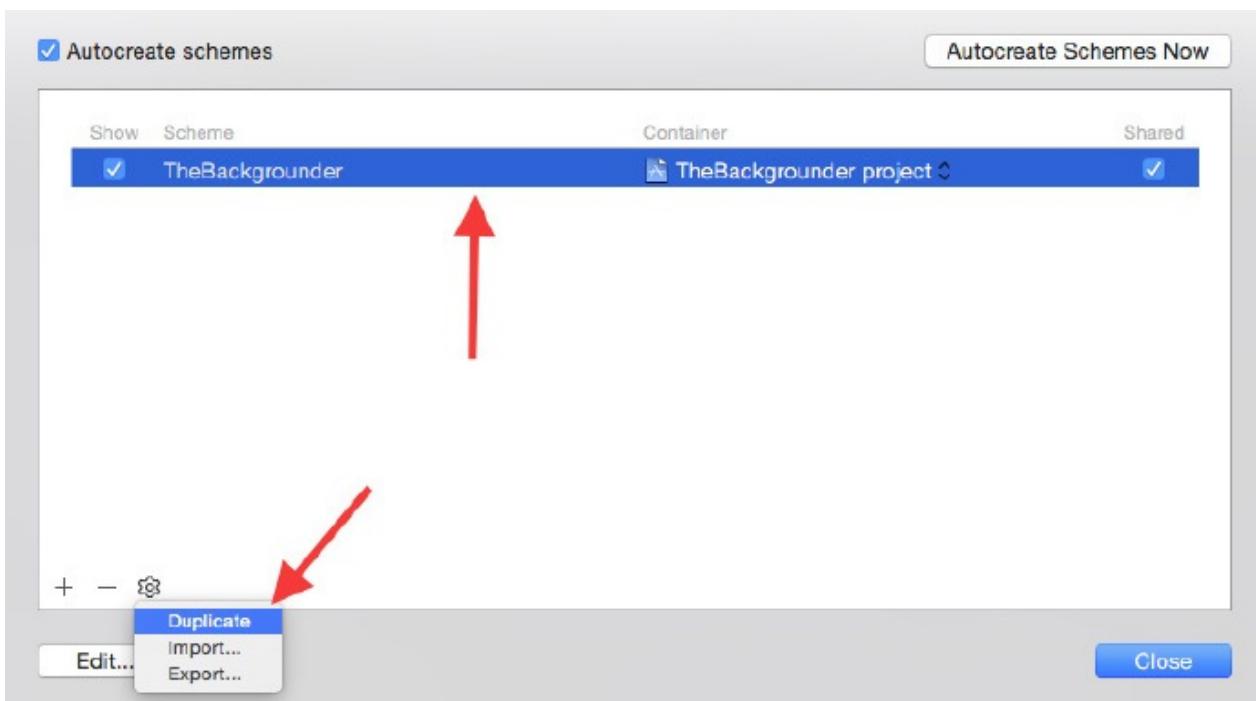
切换到Fetch选项卡,(注意当你模拟后台获取而且不是显示“Not yet updated”的时候时间)

另一种方法是在从挂起状态回复的时候测试后台获取.这里有一个启动项让你APP一运行就直接进入挂起状态.因为你可能要测试这种临界状态,用这个选项始终建立新的Scheme是最好的.Xcode使这种情况很容易实现.

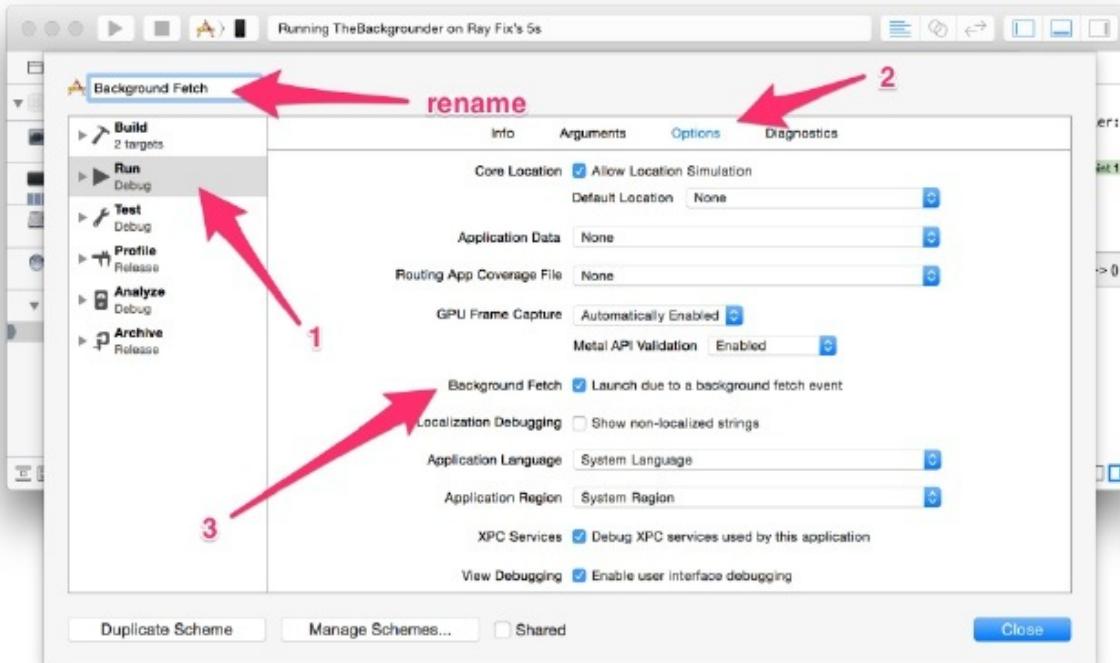
首先选择Manage Schemes选项.



接下来,选择列表里仅有的方案,然后点击齿轮图标,选择Duplicate Scheme.



最后,用合理的名字重命名你的方案,比如“Background Fetch”,并选中 Launch due to background fetch event的复选框.



需要注意的是在Xcode6.1中,在模拟器上这并不能可靠的运行.我自己测试的时候,我需要使用真正的设备正确的从启动进去到挂起状态.

用这个方案运行你的APP.你会发现,该APP没有真正的打开,而是直接运行到了挂起状态.现在,手动开启它,并进入Fetch选项.你会看到,当你运行该APP时,时间会更新,而不会显示“Not yet updated”.

使用后台获取能够有效地让你的用户们流畅的一直获取最新的内容.

## 何去何从?

[你可以在这里下载完整的示例工程.](#)

如果你想读我们这里涉及到苹果文档里的内容,最佳开始地点是 [Background Execution](#). 该文档介绍了每一个后台模式,并为每个模式链接到相应的位置.

该文档有趣的部分谈论了如何构建一个可靠的APP.你应该知道释放正在后台运行的APP中的一些细节或多或少会涉及到你到APP.

最后,如果你打算做大型网络信息传输,确保检查[NSURLSession](#).

我们希望你能享受这个课程,如果你有任何疑问或意见,请加入下面的论坛讨论.

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-25 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

专题 - 控制器介绍

## 关键字

专题 \ Controller \ 控制器

## 需求场景

1. 需要对 SDK 所提供系统控制器梳理了解时

## 参考链接

1. [Apple documentation](#)

## 详细内容

| 序号 | 控制器                                      | Class                                     | 说明              |
|----|------------------------------------------|-------------------------------------------|-----------------|
| 1  | View Controller                          | UIViewController                          | 页面视图控制器         |
| 2  | Navigation Controller                    | UINavigationController                    | 导航视图控制器         |
| 3  | Table View Controller                    | UITableViewController                     | 列表视图控制器         |
| 4  | Tab Bar Controller                       | UITabBarController                        | 标签视图控制器         |
| 5  | Split View Controller                    | UISplitViewController                     | 拆分视图控制器         |
| 6  | Page View Controller                     | UIPageViewController                      | 翻页视图控制器         |
| 7  | GLKit View Controller                    | GLKViewController                         |                 |
| 8  | Collection View Controller               | UICollectionViewController                | 集合视图展示控制器       |
| 9  | AVKit Player View Controller             | AVPlayerViewController                    | 视频播放控制器         |
| 10 | Search Bar and Search Display Controller | UISearchBar and UISearchDisplayController | 查询控制器与查询结果展示控制器 |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-28 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

专题 - 时间戳的介绍与使用

## 关键字

专题 \ Mac 系统 \ MAC OS \ 时间戳 \ Unix timestamp

## 需求场景

1. 服务器与客户端进行时间同步

## 参考链接

1. [关于Unix时间戳\(Unix timestamp\)](#)

## 详细内容

时间戳是自 1970 年 1 月 1 日（00:00:00 GMT）以来的秒数。它也被称为 Unix 时间戳（Unix Timestamp）。

Unix时间戳(Unix timestamp)，或称Unix时间(Unix time)、POSIX时间(POSIX time)，是一种时间表示方式，定义为从格林威治时间1970年01月01日00时00分00秒起至现在的总秒数。Unix时间戳不仅被使用在Unix系统、类Unix系统中，也在许多其他操作系统中被广泛采用。

如何在不同编程语言中获取现在的**Unix时间戳(Unix timestamp)**？

| 语言                        | 代码                                                                             |
|---------------------------|--------------------------------------------------------------------------------|
| Java                      | time                                                                           |
| JavaScript                | Math.round(new Date().getTime()/1000)<br>getTime()返回数值的单位是毫秒                   |
| Microsoft .NET / C#       | epoch = (DateTime.Now.ToUniversalTime().Ticks - 621355968000000000) / 10000000 |
| MySQL                     | SELECT unix_timestamp(now())                                                   |
| Perl                      | time                                                                           |
| PHP                       | time()                                                                         |
| PostgreSQL                | SELECT extract(epoch FROM now())                                               |
| Python                    | 先 import time 然后 time.time()                                                   |
| Ruby                      | 获取Unix时间戳 : Time.now 或 Time.new<br>显示Unix时间戳 : Time.now.to_i                   |
| SQL Server                | SELECT DATEDIFF(s, '1970-01-01 00:00:00', GETUTCDATE())                        |
| Unix / Linux              | date +%s                                                                       |
| VBScript / ASP            | DateDiff("s", "01/01/1970 00:00:00", Now())                                    |
| 其他操作系统<br>(如果Perl被安装在系统中) | 命令行状态 : perl -e "print time"                                                   |
| Objective-C               | [[NSDate new] timeIntervalSince1970];                                          |
| Swift                     | NSDate().timeIntervalSince1970                                                 |

如何在不同编程语言中实现**Unix时间戳(Unix timestamp)** → 普通时间 ?

| 语言                        | 代码                                                                                                                     |
|---------------------------|------------------------------------------------------------------------------------------------------------------------|
| Java                      | String date = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm:ss").format(new java.util.Date(Unix timestamp * 1000))  |
| JavaScript                | 先 var unixTimestamp = new Date(Unix timestamp * 1000) 然后 commonTime = unixTimestamp.toLocaleString()                   |
| Linux                     | date -d @Unix timestamp                                                                                                |
| MySQL                     | from_unixtime(Unix timestamp)                                                                                          |
| Perl                      | 先 my \$time = Unix timestamp 然后 my (\$sec, \$min, \$hour, \$day, \$month, \$year) = (localtime(\$time))[0,1,2,3,4,5,6] |
| PHP                       | date('r', Unix timestamp)                                                                                              |
| PostgreSQL                | SELECT TIMESTAMP WITH TIME ZONE 'epoch' + Unix timestamp) * INTERVAL '1 second';                                       |
| Python                    | 先 import time 然后 time.gmtime(Unix timestamp)                                                                           |
| Ruby                      | Time.at(Unix timestamp)                                                                                                |
| SQL Server                | DATEADD(s, Unix timestamp, '1970-01-01 00:00:00')                                                                      |
| VBScript / ASP            | DateAdd("s", Unix timestamp, "01/01/1970 00:00:00")                                                                    |
| 其他操作系统<br>(如果Perl被安装在系统中) | 命令行状态 : perl -e "print scalar(localtime(Unix timestamp))"                                                              |
| Objective-C               | [NSDate dateWithTimeIntervalSince1970:1363948516];                                                                     |
| Swift                     | NSDate(timeIntervalSince1970: 1363948516)                                                                              |

如何在不同编程语言中实现普通时间 → **Unix时间戳(Unix timestamp)** ?

| 语言             | 代码                                                                                               |
|----------------|--------------------------------------------------------------------------------------------------|
| Java           | long epoch = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm:ss").parse("01/01/1970 01:00:00"); |
| JavaScript     | var commonTime = new Date(Date.UTC(year, month - 1, day, hour, minute, second))                  |
| MySQL          | SELECT unix_timestamp(time)<br>时间格式: YYYY-MM-DD HH:MM:SS 或 YYMMDD 或 YYYYMMDD                     |
| Perl           | 先 use Time::Local 然后 my \$time = timelocal(\$sec, \$min, \$hour, \$day, \$month, \$year);        |
| PHP            | mktime(hour, minute, second, day, month, year)                                                   |
| PostgreSQL     | SELECT extract(epoch FROM date('YYYY-MM-DD HH:MM:SS'));                                          |
| Python         | 先 import time 然后 int(time.mktime(time.strptime('YYYY-MM-DD HH:MM:SS', '%Y-%m-%d %H:%M:%S')))     |
| Ruby           | Time.local(year, month, day, hour, minute, second)                                               |
| SQL Server     | SELECT DATEDIFF(s, '1970-01-01 00:00:00', time)                                                  |
| Unix / Linux   | date +%s -d"Jan 1, 1970 00:00:01"                                                                |
| VBScript / ASP | DateDiff("s", "01/01/1970 00:00:00", time)                                                       |
| Objective-C    | [date timeIntervalSince1970];                                                                    |
| Swift          | NSDate().timeIntervalSince1970                                                                   |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-12 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

二维码 - QRCode 生成与识别

## 关键字

二维码 \ QRCode \ 扫描 \ 识别

## 需求场景

1. 需要使用到二维码的扫描与识别场景

## 参考链接

1. Stack Overflow - How to use C code (libqrencode) in an iOS project
2. CSDN - iOS自带扫描 和 生成二维码
3. ITEYE - 在iOS和Android中使用二维码ZXing库及常见问题解决和整合后的代码

## 详细内容

1. ZBar
2. ZXing
3. libqrencode 生成二维码 配合 iOS 7 自带扫描QRCode

AMScanViewController.h

```
//  
//  AMScanViewController.h  
//  
//  
//  Created by Alexander Mack on 11.10.13.  
//  Copyright (c) 2013 ama-dev.com. All rights reserved.  
  
  
#import <UIKit/UIKit.h>  
#import <AVFoundation/AVFoundation.h>  
  
@protocol AMScanViewControllerDelegate;  
  
@interface AMScanViewController : UIViewController <AVCaptureMetadataOutputObjectsDelegate>  
  
@property (nonatomic, weak) id<AMScanViewControllerDelegate> delegate;  
  
@property (assign, nonatomic) BOOL touchToFocusEnabled;  
  
- (BOOL) isCameraAvailable;  
- (void) startScanning;  
- (void) stopScanning;  
- (void) setTorch:(BOOL) aStatus;  
  
@end  
  
@protocol AMScanViewControllerDelegate <NSObject>  
  
@optional  
  
- (void) scanViewController:(AMScanViewController *) aCtler didTapToFocusOnPoint:(CGPoint) aPoint;  
- (void) scanViewController:(AMScanViewController *) aCtler didSuccessfullyScan:(NSString *) aScannedValue;  
  
@end
```

## AMScanViewController.m

```
//  
//  AMScanViewController.m  
//  
//  Created by Alexander Mack on 11.10.13.  
//  Copyright (c) 2013 ama-dev.com. All rights reserved.  
  
  
#import "AMScanViewController.h"  
  
  
@interface AMScanViewController ()  
  
  
@property (strong, nonatomic) AVCaptureDevice* device;  
@property (strong, nonatomic) AVCaptureDeviceInput* input;  
@property (strong, nonatomic) AVCaptureMetadataOutput* output;  
@property (strong, nonatomic) AVCaptureSession* session;  
@property (strong, nonatomic) AVCaptureVideoPreviewLayer* preview;  
  
@end  
  
  
@implementation AMScanViewController  
  
  
- (id)initWithNibName:(NSString *)NibNameOrNil bundle:(NSBundle *)nibBundleOrNilOrNil  
{  
    self = [super initWithNibName:nibNameOrNilOrNil bundle:nibBundleOrNilOrNil];  
    if (self) {  
    }  
    return self;  
}  
  
  
- (void)viewWillAppear:(BOOL)animated;  
{  
    [super viewWillAppear:animated];  
    if (![self isCameraAvailable]) {  
        [self setupNoCameraView];  
    }  
}
```

```
    }

}

- (void) viewDidAppear:(BOOL)animated{
{
    [super viewDidAppear:animated];
}

- (void) viewDidLoad
{
    [super viewDidLoad];
    if([self isCameraAvailable]) {
        [self setupScanner];
    }
}

- (void) didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (void) touchesBegan:(NSSet*)touches withEvent:(UIEvent*)evt
{
    if(self.touchToFocusEnabled) {
        UITouch *touch=[touches anyObject];
        CGPoint pt= [touch locationInView:self.view];
        [self focus:pt];
    }
}

#pragma mark -
#pragma mark NoCamAvailable

- (void) setupNoCameraView;
{
    UILabel *labelNoCam = [[UILabel alloc] init];
    labelNoCam.text = @"No Camera available";
    labelNoCam.textColor = [UIColor blackColor];
    [self.view addSubview:labelNoCam];
```

```
[labelNoCam sizeToFit];
labelNoCam.center = self.view.center;
}

- (NSUInteger)supportedInterfaceOrientations{
{
    return UIInterfaceOrientationMaskLandscape;
}

- (BOOL)shouldAutorotate{
{
    return (UIDeviceOrientationIsLandscape([[UIDevice currentDevice] orientation]));
}

- (void)didRotateFromInterfaceOrientation:(UIInterfaceOrientation)fromInterfaceOrientation{
{
    if ([[UIDevice currentDevice] orientation] == UIDeviceOrientationLandscapeLeft) {
        AVCaptureConnection *con = self.preview.connection;
        con.videoOrientation = AVCaptureVideoOrientationLandscapeRight;
    } else {
        AVCaptureConnection *con = self.preview.connection;
        con.videoOrientation = AVCaptureVideoOrientationLandscapeLeft;
    }
}

#pragma mark -
#pragma mark AVFoundationSetup

- (void) setupScanner{
{
    self.device = [AVCaptureDevice defaultDeviceWithMediaType:AVMediaTypeVideo];

    self.input = [AVCaptureDeviceInput deviceInputWithDevice:self.device error:nil];
}
```

```

self.session = [[AVCaptureSession alloc] init];

self.output = [[AVCaptureMetadataOutput alloc] init];
[self.session addOutput:self.output];
[self.session addInput:self.input];

[self.output setMetadataObjectsDelegate:self queue:dispatch_get_main_queue()];
self.output.metadataObjectTypes = @+[AVMetadataObjectTypeQRCode];

self.preview = [AVCaptureVideoPreviewLayer layerWithSession:
self.session];
self.preview.videoGravity = AVLayerVideoGravityResizeAspectFill;
self.preview.frame = CGRectMake(0, 0, self.view.frame.size.width,
self.view.frame.size.height);

AVCaptureConnection *con = self.preview.connection;

con.videoOrientation = AVCaptureVideoOrientationLandscapeLeft
;

[self.view.layer insertSublayer:self.preview atIndex:0];
}

#pragma mark -
#pragma mark Helper Methods

- (BOOL) isCameraAvailable;
{
    NSArray *videoDevices = [AVCaptureDevice devicesWithMediaType:AVMediaTypeVideo];
    return [videoDevices count] > 0;
}

- (void)startScanning;
{
    [self.session startRunning];
}

```

```

}

- (void) stopScanning;
{
    [self.session stopRunning];
}

- (void) setTorch:(BOOL) aStatus;
{
    AVCaptureDevice *device = [AVCaptureDevice defaultDeviceWith
MediaType:AVMediaTypeVideo];
    [device lockForConfiguration:nil];
    if ( [device hasTorch] ) {
        if ( aStatus ) {
            [device setTorchMode:AVCaptureTorchModeOn];
        } else {
            [device setTorchMode:AVCaptureTorchModeOff];
        }
    }
    [device unlockForConfiguration];
}

- (void) focus:(CGPoint) aPoint;
{
    AVCaptureDevice *device = [AVCaptureDevice defaultDeviceWith
MediaType:AVMediaTypeVideo];
    if([device isFocusPointOfInterestSupported] &&
    [device isFocusModeSupported:AVCaptureFocusModeAutoFocus]
) {
        CGRect screenRect = [[UIScreen mainScreen] bounds];
        double screenWidth = screenRect.size.width;
        double screenHeight = screenRect.size.height;
        double focus_x = aPoint.x/screenWidth;
        double focus_y = aPoint.y/screenHeight;
        if([device lockForConfiguration:nil]) {
            if([self.delegate respondsToSelector:@selector(scanV
iewController:didTapToFocusOnPoint:)])
                [self.delegate scanViewController:self didTapToF
ocusOnPoint:aPoint];
        }
    }
}

```

```

    }
    [device setFocusPointOfInterest:CGPointMake(focus_x,
focus_y)];
    [device setFocusMode:AVCaptureFocusModeAutoFocus];
    if ([device isExposureModeSupported:AVCaptureExposureModeAutoExpose]){
        [device setExposureMode:AVCaptureExposureModeAutoExpose];
    }
    [device unlockForConfiguration];
}
}

#pragma mark -
#pragma mark AVCaptureMetadataOutputObjectsDelegate

- (void)captureOutput:(AVCaptureOutput *)captureOutput didOutputMetadataObjects:(NSArray *)metadataObjects
fromConnection:(AVCaptureConnection *)connection
{
    for(AVMetadataObject *current in metadataObjects) {
        if([current isKindOfClass:[AVMetadataMachineReadableCodeObject class]]) {
            if([self.delegate respondsToSelector:@selector(scanViewController:didSuccessfullyScan:)]) {
                NSString *scannedValue = [((AVMetadataMachineReadableCodeObject *) current) stringValue];
                [self.delegate scanViewController:self didSuccessfullyScan:scannedValue];
            }
        }
    }
}

@end

```

## 效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-02-04 | Alfred Jiang | -  |

## 方案名称

其他 - RESTful API 设计与最佳实践介绍

## 关键字

其他 \ RESTful API \ 接口 \ 网络

## 需求场景

1. 指定网络接口规则时

## 参考链接

(见详细内容)

## 详细内容

- [10 Best Practices for Better RESTful API](#)
- [10个有关RESTful API良好设计的最佳实践](#)
- [Best Practices for Designing a Pragmatic RESTful API](#)
- [REST API Resources\(推荐\)](#)
- [RESTFUL API 安全设计指南](#)
- [RESTful API 设计指南](#)
- [博客园 - 我所理解的RESTful Web API \[Web标准篇\]](#)
- [博客园 - 我所理解的RESTful Web API \[设计篇\]](#)
- [阮一峰的网络日志 - RESTful API 设计指南](#)
- [阮一峰的网络日志 - 理解RESTful架构](#)

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-11 | Alfred Jiang | -  |

## 方案名称

其他 - 书籍推荐

## 关键字

其他 \ 开发书籍 \ 电子书 \ pdf 免费电子书 \ 推荐

## 需求场景

1. 记录一些优秀的开发书籍

## 参考链接

1. 知乎 - 有什么iOS进阶的书推荐吗?
2. 知乎 - GitHub 上有哪些值得推荐的开源电子书?

## 详细内容

### 索引

1. [GitHub - justjavac/free-programming-books-zh\\_CN](#)
2. [GitHub - vhf/free-programming-books](#)
3. [GitHub - 免费的编程中文书籍索引](#)
4. [Justjavac - 一个合格的程序员应该读过哪些书](#)
5. [Linux Story - 编程类开放书籍荟萃](#)

## iOS 开发相关

1. 《Effective Objective-C 2.0: 编写高质量iOS与OS X代码的52个有效方法》

2. 《iOS 7 App Development Essentials》
3. 《iOS 7 by Tutorials》
4. 《iOS 7 Programming Cookbook》
5. 《iOS Auto Layout开发秘籍（第2版）》
6. 《iOS Core Animation: Advanced Techniques》
7. 《IOS数据库应用高级编程（第2版）》
8. 《iOS网络高级编程: iPhone和iPad的企业应用开发》
9. 《Learning Core Data for iOS》
10. 《Objective-C编程之道: iOS设计模式解析》
11. 《Objective-C高级编程: iOS与OS X多线程和内存管理》
12. 《Xcode 江湖录》

## 免费电子书

- 《Blocks 编程要点》
- 《Instruments 用户指南》
- 《多线程编程指南》
- 《本地和推送通知编程指南》
- 《核心动画编程指南》

## 其他

1. 《The Architecture of Open Source Applications》 :对架构感兴趣的童鞋，推荐这本，两大卷，覆盖了将近50个经典开源项目的架构，而且基本都是主力参与人员写的一线体验，直接线上可阅读
2. 《软件开发者路线图: 从学徒到高手》

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-14 | Alfred Jiang | -  |

## 方案名称

其他 - 有趣的网址收集

## 关键字

其他\有趣的网址收集\有趣的网站收集

## 需求场景

1. 打发时间

## 参考链接

(见详细内容)

## 详细内容

### 开发技术类

- [GitHub - alixaxel/ArrestDB](#):ArrestDB is a "plug-n-play" RESTful API for SQLite, MySQL and PostgreSQL databases.
- [GitHub - Alterplay/APSmartStorage](#):网络文件数据缓存库
- [GitHub - BoltsFramework/Bolts-ObjC](#):Bolts is a collection of low-level libraries designed to make developing mobile apps easier.
- [GitHub - brunow/BWStatusBarOverlay](#):一个 Status Bar 的扩展支持（点击和自定义显示）
- [GitHub - chukong/SampleGame-FantasyWarrior3D](#):A 3D sample game powered by Cocos2d-x v3.8

- [GitHub - CloudSide/VdiskSDK-iOS](#):微盘iOS客户端SDK
- [GitHub - CocoaChina-editors/Welcome-to-Swift](#):Swift 学习资源
- [GitHub - cokecoffe/ios-demo](#):demos for ios study.
- [GitHub - crosslife/OpenBird](#):利用Cocos2d-x 3.0 beta2新物理特性做出的《Flappy Bird》开源完整版
- [GitHub - dimzzy/BaseAppKit](#):代码工具集
- [GitHub - downie/Viewfinder](#):Debug 工具，点击屏幕输出页面继承层级关系
- [GitHub - edison9888/MKAnnotationViewDemo](#):地图标签簇
- [GitHub - forcedotcom/SalesforceMobileSDK-iOS](#):Salesforce.com Mobile SDK for iOS
- [GitHub - foundry/NSViewControllerPresentation](#):一个自定义 Mac 组件
- [GitHub - futurice/ios-good-practices](#):iOS 开发的一些最佳实践
- [GitHub - geekcompany/DeerResume](#):最好用的MarkDown在线简历工具，可在线预览、编辑、设置访问密码和生成PDF
- [GitHub - geekcompany/ResumeSample](#):程序员简历模板系列
- [GitHub - hilen/TSWeChat](#):用 Swift 仿写了一下微信客户端
- [GitHub - iidioter/idea](#):代码工具集
- [GitHub - iOS开发技术前线](#):开发博客
- [GitHub - M157q/shadowsocks](#):Shadowsocks
- [GitHub - marcoarment/FCUtilities](#):代码工具集
- [GitHub - MessageDisplayKit](#):仿 WeChat 客户端
- [GitHub - nixzhu/dev-blog](#):开发博客
- [GitHub - Parse Server](#):Parse 支持许多常见的后台功能，例如扩展服务器以及发送推送通知，从而降低开发者编写本地移动应用的难度。[Parse-SDK-iOS-OSX](#)
- [GitHub - processone/ejabberd](#):Robust, ubiquitous and massively scalable Jabber / XMPP Instant Messaging platform
- [GitHub - psineur/NSObject-AutomaticCoding](#):一个可以自动将序列化的对象转换为实例对象的第三方库，功能比较强大
- [GitHub - qdvictory/EasyToMore](#):打包脚本
- [GitHub - raywenderlich/SKTUtils](#):A collection of Sprite Kit helper classes and functions, written in Swift.
- [GitHub - realtimeprojects/quiaxplorer](#):Web based file management
- [GitHub - soffes/SAMCategories](#):代码工具集
- [GitHub - tanhaogg/Vdisk](#):微盘For Mac
- [GitHub - terryso/NSDate-StarSign](#):这是一个NSDate的Category，功能就是返

回日期对应的星座的字符串

- [GitHub - thoughtbot/TBAnnotationClustering](#): 地图标签簇
- [GitHub - Trinea/android-common](#): Android 开发公共组件
- [GitHub - Trinea/android-open-project](#): Android 开源项目分类汇总
- [GitHub - vhf/free-programming-books](#): 免费的编程电子书

## 计算机基础类

- [Data Structure Visualizations](#): 这是一个常见数据结构可视化的网站，非常适合初学者直观地了解数据结构的工作过程

## 设计素材类

- [Baubau Haus](#): 格调满满的图库网站
- [设计师网址导航](#): 设计素材导航
- [优设 - 你知道这些ICON宝库吗？15个免费下载图标的好去处](#)
- [Google Design - Material icons](#)

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-25 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

内购 - iOS 内购的快速实现

## 关键字

内购 \ Purchase \ 应用内购买 \ In-App Purchases

## 需求场景

1. 需要实现应用内购买需求时

## 参考链接

1. [Cocos2d应用内购买及IAP](#)
2. [GitHub - MKStoreKit](#)
3. [博客园 - In App Purchases 入门\[译\]\(原文\)](#)
4. [51CTO - iOS应用内置付费:In-App Purchases完全攻略\(1\)](#)
5. [51CTO - iOS应用内置付费:In-App Purchases完全攻略\(2\)](#)
6. [Apple documentation - In-App Purchase Programming Guide](#)
7. [MKStoreKit小记](#)

## 详细内容

### 1. 初始化

```
[[MKStoreKit sharedKit] startProductRequest];
```

## 2. 获取 IAP 列表

```
[[NSNotificationCenter defaultCenter] addObserverForName:kMKStoreKitProductsAvailableNotification
                                              object:nil
                                                queue:[[NSOperationQueue alloc] init]
                                         usingBlock:^(NSNotification *note) {
                                             NSLog(@"%@", [[MKStoreKit sharedKit] availableProducts]);
                                         }];
}
```

## 3. 购买 IAP

```
[[MKStoreKit sharedKit] initiatePaymentRequestForProductWithIdentifier:productIdentifier];

[[NSNotificationCenter defaultCenter] addObserverForName:kMKStoreKitProductPurchasedNotification
                                              object:nil
                                                queue:[[NSOperationQueue alloc] init]
                                         usingBlock:^(NSNotification *note) {
                                             NSLog(@"Purchased/Subscribed to product with id: %@", [note object]);
                                             NSLog(@"%@", [[MKStoreKit sharedKit] valueForKey:@"purchaseRecord"]);
                                         }];
}
```

## 4. 恢复 IAP

```
[[NSNotificationCenter defaultCenter] addObserverForName:kMKStoreKitRestoredPurchasesNotification
                                              object:nil
                                                queue:[[NSOperationQueue alloc] init]
                                         usingBlock:^(NSNotification *note) {
                                             NSLog(@"%@", note);
                                         }];

[[NSNotificationCenter defaultCenter] addObserverForName:kMKStoreKitRestoringPurchasesFailedNotification
                                              object:nil
                                                queue:[[NSOperationQueue alloc] init]
                                         usingBlock:^(NSNotification *note) {
                                             NSLog(@"Failed restoring purchases with error: %@", note.object);
                                         }];
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-04 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

动画 - Core Animation 之 CABaseAnimation

## 关键字

动画 \ Core Animation \ CALayer \ CAAnimation \ CAPropertyAnimation \ CABaseAnimation \ CAKeyframeAnimation \ CATransition \ CAAnimationGroup

## 需求场景

1. 使用 Core Animation 实现动画需求

## 参考链接

1. CSDN - CABasicAnimation的基本使用方法（移动·旋转·放大·缩小）
2. 博客园 - CABasicAnimation animationWithKeyPath 一些规定的值
3. 简书 - 干货系列之手把手教你使用Core animation 做动画 (推荐)
4. 《核心动画编程指南》

## 详细内容

### 1. 继承关系

```

@interface CAAnimation : NSObject <NSCoding, NSCopying, CAMediaTiming, CAAction>

@interface CAPropertyAnimation : CAAnimation

@interface CABasicAnimation : CAPropertyAnimation

@interface CAKeyframeAnimation : CAPropertyAnimation

@interface CATransition : CAAnimation

@interface CAAnimationGroup : CAAnimation

```

## 2. 使用说明

### 1. 定义一个 CABasicAnimation

```

// 指定position属性
CABasicAnimation *animation = [CABasicAnimation animationWithKeyPath:@"position"];

```

### 2. 设定动画属性

| 属性             | 说明                                                        |
|----------------|-----------------------------------------------------------|
| duration       | 动画时长（秒为单位）                                                |
| repeatCount    | 重复次数。永久重复的话设置为HUGE_VALF                                   |
| beginTime      | 指定动画开始时间。从开始指定延迟几秒执行的话，请设置为「CACurrentMediaTime() + 秒数」的形式 |
| timingFunction | 设定动画的速度变化                                                 |
| autoreverses   | 动画结束时是否执行逆动画                                              |

```

animation.duration = 2.5; // 动画持续时间
animation.repeatCount = 1; // 不重复
animation.beginTime = CACurrentMediaTime() + 2; // 2秒后执行
animation.autoreverses = YES; // 结束后执行逆动画

// 动画先加速后减速
animation.timingFunction = [CAMediaTimingFunction functionWithName: kCAMediaTimingFunctionEaseInEaseOut];

```

### 3. 设定动画的开始帧和结束帧

设定动画开始和结束帧时的状态。设定的值会变为 *KeyPath* 所指定的属性的值。

| 属性        | 说明       |
|-----------|----------|
| fromValue | 开始值      |
| toValue   | 终了值（绝对值） |
| byValue   | 终了值（相对值） |

```

// 指定position属性（移动）
CABasicAnimation *animation =
    [CABasicAnimation animationWithKeyPath:@"position"];

. . .

// 设定动画起始帧和结束帧
animation.fromValue = [NSValue valueWithCGPoint:CGPointMake(0, 0)];
// 起始点
animation.toValue = [NSValue valueWithCGPoint:CGPointMake(320, 480)];
// 终了点

```

### 4. 添加动画

```
[myView.layer addAnimation:animation forKey:@"move-layer"];
```

### 5. 动画结束恢复初始状态（可选）

```
// 动画终了后不返回初始状态  
animation.removedOnCompletion = NO;  
animation.fillMode = kCAFillModeForwards;
```

### 2. 使用举例

#### 1. 移动动画

```
/* 移动 */  
CABasicAnimation *animation = [CABasicAnimation animationWithKey  
Path:@"position"];  
  
// 动画选项的设定  
animation.duration = 2.5; // 持续时间  
animation.repeatCount = 1; // 重复次数  
  
// 起始帧和终了帧的设定  
animation.fromValue = [NSValue valueWithCGPoint:myView.layer.pos  
ition]; // 起始帧  
animation.toValue = [NSValue valueWithCGPoint:CGPointMake(320, 4  
80)]; // 终了帧  
  
// 添加动画  
[myView.layer addAnimation:animation forKey:@"move-layer"];
```

#### 2. 旋转动画

```
/* 旋转 */  
  
// 对Y轴进行旋转（指定Z轴的话，就和UIView的动画一样绕中心旋转）  
CABasicAnimation *animation = [CABasicAnimation animationWithKey  
Path:@"transform.rotation.y"];  
  
// 设定动画选项  
animation.duration = 2.5; // 持续时间  
animation.repeatCount = 1; // 重复次数  
  
// 设定旋转角度  
animation.fromValue = [NSNumber numberWithFloat:0.0]; // 起始角度  
animation.toValue = [NSNumber numberWithFloat:2 * M_PI]; // 终止  
角度  
  
// 添加动画  
[myView.layer addAnimation:animation forKey:@"rotate-layer"];
```

### 3. 缩放动画

```
/* 放大缩小 */  
  
// 设定为缩放  
CABasicAnimation *animation = [CABasicAnimation animationWithKey  
Path:@"transform.scale"];  
  
// 动画选项设定  
animation.duration = 2.5; // 动画持续时间  
animation.repeatCount = 1; // 重复次数  
animation.autoreverses = YES; // 动画结束时执行逆动画  
  
// 缩放倍数  
animation.fromValue = [NSNumber numberWithFloat:1.0]; // 开始时的  
倍率  
animation.toValue = [NSNumber numberWithFloat:2.0]; // 结束时的倍率  
  
// 添加动画  
[myView.layer addAnimation:animation forKey:@"scale-layer"];
```

#### 4. 组合动画

```
/* 动画1（在X轴方向移动） */
CABasicAnimation *animation1 =
    [CABasicAnimation animationWithKeyPath:@"transform.translation.x"];
// 终点设定
animation1.toValue = [NSNumber numberWithFloat:80]; // 终点

/* 动画2（绕Z轴中心旋转） */
CABasicAnimation *animation2 =
    [CABasicAnimation animationWithKeyPath:@"transform.rotation.z"];
// 设定旋转角度
animation2.fromValue = [NSNumber numberWithFloat:0.0]; // 开始时的角度
animation2.toValue = [NSNumber numberWithFloat:4 * M_PI]; // 结束时的角度

/* 动画组 */
CAAnimationGroup *group = [CAAnimationGroup animation];

// 动画选项设定
group.duration = 3.0;
group.repeatCount = 1;

// 添加动画
group.animations = [NSArray arrayWithObjects:animation1, animation2, nil];
[myView.layer addAnimation:group forKey:@"move-rotate-layer"];
```

## 5. 捕获动画开始时和终了时的事件

```

/* 移动 */
CABasicAnimation *animation = [CABasicAnimation animationWithKey
Path:@"transform.translation.y"];
animation.delegate = self; // 指定委托对象

// 设定动画选项
animation.duration = 2.5; // 动画时长
animation.repeatCount = 1; // 重复次数

// 终点设定
animation.toValue = [NSNumber numberWithFloat:100];; // 终点

// 添加动画
[myView.layer addAnimation:animation forKey:@"move-layer"];

. . .

/***
 * 动画开始时
 */
- (void)animationDidStart:(CAAnimation *)theAnimation
{
    NSLog(@"begin");
}

/***
 * 动画结束时
 */
- (void)animationDidStop:(CAAnimation *)theAnimation finished:(BOOL)flag
{
    NSLog(@"end");
}

```

## 6. 几个可以用来实现热门APP应用PATH中menu效果的几个方法

```

+(CABasicAnimation *)opacityForever_Animation:(float)time //永
久闪烁的动画
{

```

```
CABasicAnimation *animation=[CABasicAnimation animationWithKeyPath:@"opacity"];
    animation.fromValue=[NSNumber numberWithFloat:1.0];
    animation.toValue=[NSNumber numberWithFloat:0.0];
    animation.autoreverses=YES;
    animation.duration=time;
    animation.repeatCount=FLT_MAX;
    animation.removedOnCompletion=NO;
    animation.fillMode=kCAFillModeForwards;
    return animation;
}

+(CABasicAnimation *)opacityTimes_Animation:(float)repeatTimes durationTimes:(float)time; //有闪烁次数的动画
{
    CABasicAnimation *animation=[CABasicAnimation animationWithKeyPath:@"opacity"];
    animation.fromValue=[NSNumber numberWithFloat:1.0];
    animation.toValue=[NSNumber numberWithFloat:0.4];
    animation.repeatCount=repeatTimes;
    animation.duration=time;
    animation.removedOnCompletion=NO;
    animation.fillMode=kCAFillModeForwards;
    animation.timingFunction=[CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseIn];
    animation.autoreverses=YES;
    return animation;
}

+(CABasicAnimation *)moveX:(float)time X:(NSNumber *)x //横向移动
{
    CABasicAnimation *animation=[CABasicAnimation animationWithKeyPath:@"transform.translation.x"];
    animation.toValue=x;
    animation.duration=time;
    animation.removedOnCompletion=NO;
    animation.fillMode=kCAFillModeForwards;
    return animation;
}
```

```

+(CABasicAnimation *)moveY:(float)time Y:(NSNumber *)y //纵向移
动
{
    CABasicAnimation *animation=[CABasicAnimation animationWithK
eyPath:@"transform.translation.y"];
    animation.toValue=y;
    animation.duration=time;
    animation.removedOnCompletion=NO;
    animation.fillMode=kCAFillModeForwards;
    return animation;
}

+(CABasicAnimation *)scale:(NSNumber *)Multiple
                    origin:(NSNumber *)originMultiple
durTimes:(float)time
Rep:(float)repeatTimes //缩放
{
    CABasicAnimation *animation=[CABasicAnimation animationWithK
eyPath:@"transform.scale"];
    animation.fromValue=originMultiple;
    animation.toValue=Multiple;
    animation.duration=time;
    animation.autoreverses=YES;
    animation.repeatCount=repeatTimes;
    animation.removedOnCompletion=NO;
    animation.fillMode=kCAFillModeForwards;
    return animation;
}

+(CAAnimationGroup *)groupAnimation:(NSArray *)animationAry
                                durTimes:(float)time
                                Rep:(float)repeatTimes //组合动
画
{
    CAAnimationGroup *animation=[CAAnimationGroup animation];
    animation.animations=animationAry;
    animation.duration=time;
    animation.repeatCount=repeatTimes;
    animation.removedOnCompletion=NO;
}

```

```

        animation.fillMode=kCAFillModeForwards;
        return animation;
    }

+(CAKeyframeAnimation *)keyframeAniamtion:(CGMutablePathRef)path
                                    durTimes:(float)time
                                    Rep:(float)repeatTimes ///
路径动画
{
    CAKeyframeAnimation *animation=[CAKeyframeAnimation animation
nWithKeyPath:@"position"];
    animation.path=path;
    animation.removedOnCompletion=NO;
    animation.fillMode=kCAFillModeForwards;
    animation.timingFunction=[CAMediaTimingFunction functionWith
Name:kCAMediaTimingFunctionEaseIn];
    animation.autoreverses=NO;
    animation.duration=time;
    animation.repeatCount=repeatTimes;
    return animation;
}

+(CABasicAnimation *)movepoint:(CGPoint )point //点移动
{
    CABasicAnimation *animation=[CABasicAnimation animationWithK
eyPath:@"transform.translation"];
    animation.toValue=[NSValue valueWithCGPoint:point];
    animation.removedOnCompletion=NO;
    animation.fillMode=kCAFillModeForwards;
    return animation;
}

+(CABasicAnimation *)rotation:(float)dur
                           degree:(float)degree
                           direction:(int)direction
                           repeatCount:(int)repeatCount //旋转
{
    CATransform3D rotationTransform = CATransform3DMakeRotation
(degree, 0, 0,direction);
    CABasicAnimation* animation;
}

```

```
animation = [CABasicAnimation animationWithKeyPath:@"transform"];
animation.toValue= [NSValue valueWithCATransform3D:rotationTransform];
animation.duration= dur;
animation.autoreverses= NO;
animation.cumulative= YES;
animation.removedOnCompletion=NO;
animation.fillMode=kCAFillModeForwards;
animation.repeatCount= repeatCount;
animation.delegate= self;
return animation;
}
```

## 效果图

(无)

## 备注

- 动画 - Core Animation 之 CABaseAnimation
- 动画 - Core Animation 之 CATransform3D
- 动画 - Core Animation 之 Key Path

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

动画 - Core Animation 之 CATransform3D

## 关键字

动画 \ Core Animation \ CALayer \ CATransform3D

## 需求场景

1. 使用 Core Animation 实现动画需求

## 参考链接

1. CSDN - iPhone/iOS Core Animation开发总结(CALayer)
2. Sina - CATransform3D 特效详解
3. Core Animation编程指南(三)几何变换
4. 博客园 - Core Animation编程指南(译文) (推荐)
5. Apple documentation - Core Animation Programming Guide(英文好的推荐)
6. 简书 - 干货系列之手把手教你使用Core animation 做动画 (推荐)
7. 《核心动画编程指南》

## 详细内容

### 1. 使用CATransform3D函数

表 1 CATransform3D 变换函数 : 偏移、旋转和缩放

| Function                     | Use                                                                                                                                                 |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| CATransform3DMakeTranslation | Returns a transform that translates by '(tx, ty, tz)'. $t' = [1 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0; 0 \ 0 \ 1 \ 0; tx \ ty \ tz \ 1]$ .                     |
| CATransform3DTranslate       | Translate 't' by '(tx, ty, tz)' and return the result: $t' = \text{translate}(tx, ty, tz) t$ .                                                      |
| CATransform3DMakeScale       | Returns a transform that scales by '(sx, sy, sz)': $* t' = [sx \ 0 \ 0 \ 0; 0 \ sy \ 0 \ 0; 0 \ 0 \ sz \ 0; 0 \ 0 \ 1]$ .                           |
| CATransform3DScale           | Scale 't' by '(sx, sy, sz)' and return the result: $t' = \text{scale}(sx, sy, sz) t$ .                                                              |
| CATransform3DMakeRotation    | Returns a transform that rotates by 'angle' radians about the vector '(x, y, z)'. If the vector has length zero the identity transform is returned. |
| CATransform3DRotate          | Rotate 't' by 'angle' radians about the vector '(x, y, z)' and return the result. $t' = \text{rotation}(angle, x, y, z) * t$ .                      |

表 2 CATransform3D 与 CGAffineTransform 转换

| Function                         | Use                                                                                    |
|----------------------------------|----------------------------------------------------------------------------------------|
| CATransform3DMakeAffineTransform | Returns a CATransform3D with the same effect as the passed affine transform.           |
| CATransform3DIsAffine            | Returns YES if the passedCATransform3D can be exactly represented an affine transform. |
| CATransform3DGetAffineTransform  | Returns the affine transform represented by the passedCATransform3D.                   |

表 3 CATransform3D 相等测试

| Function                      | Use                                                     |
|-------------------------------|---------------------------------------------------------|
| CATransform3DIsIdentity       | Returns YES if the transform is the identity transform. |
| CATransform3DEqualToTransform | Returns YES if the two transforms are exactly equal.    |

```
[UIView animateWithDuration:0.5 animations:^{
    //    sender.layer.transform = CATransform3DMakeAffineTransform(
    CGAffineTransformMakeScale(1, 1)); //原位置
    //    sender.layer.transform = CATransform3DMakeAffineTransform(
    CGAffineTransformMakeScale(1, -1)); //上下颠倒
    //    sender.layer.transform = CATransform3DMakeAffineTransform(
    CGAffineTransformMakeScale(-1, 1)); //左右颠倒
    //    sender.layer.transform = CATransform3DMakeAffineTransform(
    CGAffineTransformMakeScale(-1, -1)); //上下左右颠倒
    //    //CATransform3D系列方法
    //    sender.layer.transform = CATransform3DMakeScale(1, 1,
    1);
    //    sender.layer.transform = CATransform3DMakeScale(-1, 1,
    1); //Y轴颠倒
    //    sender.layer.transform = CATransform3DMakeScale(1, -1,
    1); //X轴颠倒
    //    sender.layer.transform = CATransform3DMakeScale(1, 1,
    -1); //Z轴颠倒
    //    sender.layer.transform = CATransform3DScale(CATransfor
    m3DMakeScale(2, 2, 2), 1, -1, 1); //XYZ放大到2倍，再X轴颠倒
    sender.layer.transform = CATransform3DIdentity; //恢复最
初状态 如果不恢复初始状态可能导致按钮无法点击
}];
```

## 2. 直接修改数据结构的成员

```
struct CATransform3D
{
    CGFloat m11, m12, m13, m14;
    CGFloat m21, m22, m23, m24;
    CGFloat m31, m32, m33, m34;
    CGFloat m41, m42, m43, m44;
};
```

| 恒等变换                                                                                                                                      | 平移                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$                                          | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ tx & ty & tz & 1 \end{bmatrix}$                                       |
| 缩放                                                                                                                                        | 绕X轴旋转                                                                                                                                     |
| $\begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$                                       | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| 绕Y轴旋转                                                                                                                                     | 绕Z轴旋转                                                                                                                                     |
| $\begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |

### 3. 通过键值路径修改变换

表 4 CATransform3D key paths

| Field Key Path | Description                                                                                 |
|----------------|---------------------------------------------------------------------------------------------|
| rotation.x     | The rotation, in radians, in the x axis.                                                    |
| rotation.y     | The rotation, in radians, in the y axis.                                                    |
| rotation.z     | The rotation, in radians, in the z axis.                                                    |
| rotation       | The rotation, in radians, in the z axis. This is identical to setting the rotation.z field. |
| scale.x        | Scale factor for the x axis.                                                                |
| scale.y        | Scale factor for the y axis.                                                                |
| scale.z        | Scale factor for the z axis.                                                                |
| scale          | Average of all three scale factors.                                                         |
| translation.x  | Translate in the x axis.                                                                    |
| translation.y  | Translate in the y axis.                                                                    |
| translation.z  | Translate in the z axis.                                                                    |
| translation    | Translate in the x and y axis. Value is an NSSize or CGSize.                                |

你不可通过Objective-C 2.0的属性来设置结构域的值，比如下面的代码将会无法正常运行：

```
myLayer.transform.rotation.x=0;
```

替换的办法是，你必须通过setValue:forKeyPath:或者valueForKeyPath:方法，具体如下：

```
[myLayer setValue:[NSNumber numberWithInt:0] forKeyPath:@"transform.rotation.x"];
```

## 效果图

(无)

## 备注

- 动画 - Core Animation 之 CABaseAnimation
- 动画 - Core Animation 之 CATransform3D
- 动画 - Core Animation 之 Key Path

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-04 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

动画 - Core Animation 之 Key Path

## 关键字

动画 \ Core Animation \ CAAnimation \ CALayer

## 需求场景

1. CAAnimation 、 CALayer 实现动画需求时 setValue:forKeyPath: 和 valueForKeyPath: 支持的 Key Path 速查

## 参考链接

1. Apple documentation - Core Animation Programming Guide
2. 简书 - 干货系列之手把手教你使用Core animation 做动画 (推荐)
3. 《核心动画编程指南》

## 详细内容

表 1 支持的 Key Path 类型

| C type            | Wrapping class                |
|-------------------|-------------------------------|
| CGPoint           | NSValue                       |
| CGSize            | NSValue                       |
| CGRect            | NSValue                       |
| CATransform3D     | NSValue                       |
| CGAffineTransform | NSAffineTransform (OS X only) |

表 2 CATransform3D key paths

| Field Key Path | Description                                                                                 |
|----------------|---------------------------------------------------------------------------------------------|
| rotation.x     | The rotation, in radians, in the x axis.                                                    |
| rotation.y     | The rotation, in radians, in the y axis.                                                    |
| rotation.z     | The rotation, in radians, in the z axis.                                                    |
| rotation       | The rotation, in radians, in the z axis. This is identical to setting the rotation.z field. |
| scale.x        | Scale factor for the x axis.                                                                |
| scale.y        | Scale factor for the y axis.                                                                |
| scale.z        | Scale factor for the z axis.                                                                |
| scale          | Average of all three scale factors.                                                         |
| translation.x  | Translate in the x axis.                                                                    |
| translation.y  | Translate in the y axis.                                                                    |
| translation.z  | Translate in the z axis.                                                                    |
| translation    | Translate in the x and y axis. Value is an NSSize or CGSize.                                |

```
[myLayer setValue:[NSNumber numberWithFloat:10.0] forKeyPath:  
@"transform.translation.x"];
```

表 3 CGPoint Key Paths

| Structure Field | Description                   |
|-----------------|-------------------------------|
| x               | The x component of the point. |
| y               | The y component of the point. |

表 4 CGSize Key Paths

| Structure Field | Description                       |
|-----------------|-----------------------------------|
| width           | The width component of the size.  |
| height          | The height component of the size. |

表 5 CGRect Key Paths

| Structure Field | Description                                 |
|-----------------|---------------------------------------------|
| origin          | The origin of the rectangle as a CGPoint.   |
| origin.x        | The x component of the rectangle origin.    |
| origin.y        | The y component of the rectangle origin.    |
| size            | The size of the rectangle as a CGSize.      |
| size.width      | The width component of the rectangle size.  |
| size.height     | The height component of the rectangle size. |

## 效果图

(无)

## 备注

- 动画 - Core Animation 之 CABaseAnimation
- 动画 - Core Animation 之 CATransform3D
- 动画 - Core Animation 之 Key Path

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-03 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

动画 - 使用 POViewFrameBuilder 快速实现 UIView 的动画移动和布局

## 关键字

动画 \ UIView \ Layout \ Animation

## 需求场景

1. 动画实现子 UIView 的布局动画和动画移动效果

## 参考链接

1. [GitHub - POViewFrameBuilder](#)

## 详细内容

1. 将 *POViewFrameBuilder* 文件加入工程
2. 引入 *UIView+POViewFrameBuilder.h* 头文件
3. 使用示例 ``objectivec //Resizing a view: [view.po\_frameBuilder  
setWidth:100.0f height:40.0f];

```
//Moving a view to be centered within it's superview: [view.po_frameBuilder  
centerInSuperview];
```

```
//You can combine these methods to your own liking: [[view.po_frameBuilder  
setWidth:100.0f height:40.0f] centerHorizontallyInSuperview]; ``
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-03 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

动画 - 页面跳转 - 自定义模态跳转动画

## 关键字

动画 \ 跳转 \ UINavigationController \ UINavigationControllerDelegate \  
UIViewControllerAnimatedTransitioning

## 需求场景

1. 对页面跳转动画有特殊需求时

## 参考链接

1. [How To Make A View Controller Transition Animation Like in the Ping App](#)

## 详细内容

### UINavigationController

1. Swift 解决方案

定义 **NavigationControllerDelegate.swift**

```
//  
//  NavigationControllerDelegate.swift  
//  CircleTransition  
//
```

```
// Created by Rounak Jain on 23/10/14.  
// Copyright (c) 2014 Rounak Jain. All rights reserved.  
  
import UIKit  
  
class NavigationControllerDelegate: NSObject, UINavigationControllerDelegate {  
  
    var startPoint : CGPoint?  
    weak var navigationController: UINavigationController?  
    var interactionController: UIPercentDrivenInteractiveTransition?  
  
    override func awakeFromNib() {  
        super.awakeFromNib()  
        var panGesture = UIPanGestureRecognizer(target: self, action: Selector("panned:"))  
        self.navigationController!.view.addGestureRecognizer(panGesture)  
    }  
  
    func panned(gestureRecognizer: UIPanGestureRecognizer) {  
  
        switch gestureRecognizer.state {  
        case .Began:  
  
            startPoint = gestureRecognizer.locationInView(self.navigationController?.view)  
  
            if startPoint!.x < 20  
            {  
                self.interactionController = UIPercentDrivenInteractiveTransition()  
                if self.navigationController?.viewControllers.count > 1 {  
                    GJViewTools.CLICK_POINT = gestureRecognizer.locationInView(self.navigationController?.view)  
                    GJViewTools.DRAG_POINT = true  
                    self.navigationController?.popViewControllerAnimated(true)  
                }  
            }  
        }  
    }  
}
```

```

Animated(true)
    }
}

case .Changed:

    if startPoint!.x < 20
    {
        var translation = gestureRecognizer.translationInView(self.navigationController!.view)
        var completionProgress = translation.x/CGRectGetWidth(self.navigationController!.view.bounds)
        self.interactionController?.updateInteractiveTransition(completionProgress)
    }

case .Ended:

    if startPoint!.x < 20
    {

        if (gestureRecognizer.velocityInView(self.navigationController!.view).x > 0) {
            self.interactionController?.finishInteractiveTransition()
        } else {
            self.interactionController?.cancelInteractiveTransition()
        }
        self.interactionController = nil
    }

default:
    self.interactionController?.cancelInteractiveTransition()
    self.interactionController = nil
}

func navigationController(navigationController: UINavigationController

```

```

Controller, animationControllerForOperation operation: UINavigationControllerOperation, fromViewController fromVC: UIViewController, toViewController toVC: UIViewController) -> UIViewController
rAnimatedTransitioning? {
    var cirAnimator : CircleTransitionAnimator = CircleTransitionAnimator()
    cirAnimator.navigationOperation = operation
    return cirAnimator
}

func navigationController(navigationController: UINavigationController, interactionControllerForAnimationController animationController: UIViewControllerAnimatedTransitioning) -> UIViewControllerInteractiveTransitioning? {
    return self.interactionController
}
}

```

## CircleTransitionAnimator.swift

```

//
// CircleTransitionAnimator.swift
// CircleTransition
//
// Created by Rounak Jain on 23/10/14.
// Copyright (c) 2014 Rounak Jain. All rights reserved.
//


import UIKit

class CircleTransitionAnimator: NSObject, UIViewControllerAnimatedTransitioning {

    weak var transitionContext: UIViewControllerContextTransitioning?
    var navigationOperation: UINavigationControllerOperation?

    func transitionDuration(transitionContext: UIViewControllerContextTransitioning) -> NSTimeInterval {

```

```

        return GJViewTools.DURATION_TIME;
    }

    func animateTransition(transitionContext: UIViewControllerContextTransitioning) {
        self.transitionContext = transitionContext

        var containerView = transitionContext.containerView()
        var fromViewController : UIViewController = transitionContext.viewControllerForKey(UIViewContextFromViewControllerKey)!

        var toViewController : UIViewController = transitionContext.viewControllerForKey(UIViewContextToViewControllerKey)!

        var button = UIButton(frame: CGRectMake(0, 0, 10, 10))

        button.center = GJViewTools.CLICK_POINT

        if navigationOperation == UINavigationControllerOperation.Pop && !GJViewTools.DRAG_POINT {

            var snapshotImageView: UIView = fromViewController.view.snapshotViewAfterScreenUpdates(false)
            toViewController.view.addSubview(snapshotImageView)
            containerView.addSubview(toViewController.view)

            var circleMaskPathInitial = UIBezierPath(ovalInRect: button.frame)
            var radius = sqrt((SCREEN_HEIGHT*SCREEN_HEIGHT) + (SCREEN_WIDTH*SCREEN_WIDTH))
            var circleMaskPathFinal = UIBezierPath(ovalInRect: CGRectInset(button.frame, -radius, -radius))

            var maskLayer = CAShapeLayer()
            maskLayer.path = circleMaskPathFinal.CGPath
            snapshotImageView.layer.mask = maskLayer

            var maskLayerAnimation = CABasicAnimation(keyPath: "path")
            maskLayerAnimation.fromValue = circleMaskPathFinal.C

```

```

GPath
    maskLayerAnimation.toValue = circleMaskPathInitial.CGPath
GPath
    maskLayerAnimation.duration = self.transitionDuration(transitionContext)
    maskLayerAnimation.delegate = self
    maskLayer.addAnimation(maskLayerAnimation, forKey: "path")
}

self.delay(self.transitionDuration(transitionContext) - 0.1, closure: { () -> () in
    snapshotImageView.removeFromSuperview()
})
}
else
{
    containerView.addSubview(toViewController.view)

    var circleMaskPathInitial = UIBezierPath(ovalInRect: button.frame)
    var radius = sqrt((SCREEN_HEIGHT*SCREEN_HEIGHT) + (SCREEN_WIDTH*SCREEN_WIDTH))
    var circleMaskPathFinal = UIBezierPath(ovalInRect: CGRectInset(button.frame, -radius, -radius))

    var maskLayer = CAShapeLayer()
    maskLayer.path = circleMaskPathFinal.CGPath
    toViewController.view.layer.mask = maskLayer

    var maskLayerAnimation = CABasicAnimation(keyPath: "path")
    maskLayerAnimation.fromValue = circleMaskPathInitial.CGPath
    maskLayerAnimation.toValue = circleMaskPathFinal.CGPath
    maskLayerAnimation.duration = self.transitionDuration(transitionContext)
    maskLayerAnimation.delegate = self
    maskLayer.addAnimation(maskLayerAnimation, forKey: "path")
}

```

```
    }

    override func animationDidStop(anim: CAAnimation!, finished
flag: Bool) {
    self.transitionContext?.completeTransition(!self.transit
ionContext!.transitionWasCancelled())
    self.transitionContext?.viewControllerForKey(UTTransitio
nContextFromViewControllerKey)?.view.layer.mask = nil
}

}
```

使用

```
var navDelegate : NavigationControllerDelegate = NavigationContr
ollerDelegate()

navDelegate.navigationController = frontNav
frontNav.delegate = navDelegate
navDelegate.awakeFromNib()
```

## ModelTransition

### 1. Swift 解决方案

定义 **ModelTransitionDelegate.swift**

```
//
//  ModelTransitionDelegate.swift
//  GrandJustice
//
//  Created by Alfred Jiang on 2/6/15.
//  Copyright (c) 2015 FYH. All rights reserved.
//

import UIKit

enum ModalPresentingType {
```

```

    case Present, Dismiss
}

class ModelTransitionDelegate: NSObject, UIViewControllerTransiti
oningDelegate {

    func animationControllerForPresentedController(presented: UI
ViewController, presentingController presenting: UIViewController
, sourceController source: UIViewController) -> UIViewController
rAnimatedTransitioning? {

        var modelAnimator : ModelTransitionAnimator = ModelTrans
itionAnimator()
        modelAnimator.modalPresentingType = ModalPresentingType.
Present

        return modelAnimator
    }

    func animationControllerForDismissedController(dismissed: UI
ViewController) -> UIViewControllerAnimatedTransitioning? {

        var modelAnimator : ModelTransitionAnimator = ModelTrans
itionAnimator()
        modelAnimator.modalPresentingType = ModalPresentingType.
Dismiss

        return modelAnimator
    }

}

```

## ModelTransitionAnimator.swift

```

// 
//  ModelTransitionAnimator.swift
//  GrandJustice
// 

```

```
// Created by Alfred Jiang on 2/6/15.  
// Copyright (c) 2015 FYH. All rights reserved.  
  
import UIKit  
  
class ModelTransitionAnimator: NSObject, UIViewControllerAnimatedTransitioning {  
  
    var modalPresentingType: ModalPresentingType?  
    weak var transitionContext: UIViewControllerContextTransitioning?  
  
    func transitionDuration(transitionContext: UIViewControllerContextTransitioning) -> NSTimeInterval {  
        return 0.6  
    }  
  
    func animateTransition(transitionContext: UIViewControllerContextTransitioning) {  
        self.transitionContext = transitionContext  
  
        var containerView = transitionContext.containerView()  
        var fromViewController : UIViewController = transitionContext.viewControllerForKey( UITransitionContextFromViewControllerKey )!  
        var toViewController : UIViewController = transitionContext.viewControllerForKey( UITransitionContextToViewControllerKey )!  
  
        var button = UIButton(frame: CGRectMake(0, 0, 10, 10))  
  
        button.center = GJViewTools.CLICK_POINT  
  
        if modalPresentingType == ModalPresentingType.Dismiss {  
  
            var snapshotImageView: UIImageView = fromViewController.view.snapshotViewAfterScreenUpdates(false)  
            toViewController.view.addSubview(snapshotImageView)  
            containerView.addSubview(toViewController.view)  
        }  
    }  
}
```

```

        var circleMaskPathInitial = UIBezierPath(ovalInRect:
button.frame)

        var radius = sqrt((SCREEN_HEIGH*SCREEN_HEIGH) + (SCR
EEN_WIDTH*SCREEN_WIDTH))

        var circleMaskPathFinal = UIBezierPath(ovalInRect: C
GRectInset(button.frame, -radius, -radius))

        var maskLayer = CAShapeLayer()
maskLayer.path = circleMaskPathFinal.CGPath
snapshotImageView.layer.mask = maskLayer

        var maskLayerAnimation = CABasicAnimation(keyPath: "path")
maskLayerAnimation.fromValue = circleMaskPathFinal.C
GPath
maskLayerAnimation.toValue = circleMaskPathInitial.C
GPath
maskLayerAnimation.duration = self.transitionDuratio
n(transitionContext)
maskLayerAnimation.delegate = self
maskLayer.addAnimation(maskLayerAnimation, forKey: "path")

        self.delay(self.transitionDuration(transitionContext
) - 0.1, closure: { () -> () in
snapshotImageView.removeFromSuperview()
})
}
else
{
    containerView.addSubview(toViewController.view)

        var circleMaskPathInitial = UIBezierPath(ovalInRect:
button.frame)

        var radius = sqrt((SCREEN_HEIGH*SCREEN_HEIGH) + (SCR
EEN_WIDTH*SCREEN_WIDTH))

        var circleMaskPathFinal = UIBezierPath(ovalInRect: C
GRectInset(button.frame, -radius, -radius))

        var maskLayer = CAShapeLayer()

```

```

        maskLayer.path = circleMaskPathFinal.CGPath
        toViewController.view.layer.mask = maskLayer

        var maskLayerAnimation = CABasicAnimation(keyPath: "path")
        maskLayerAnimation.fromValue = circleMaskPathInitial.CGPath
        maskLayerAnimation.toValue = circleMaskPathFinal.CGPath
        maskLayerAnimation.duration = self.transitionDuration(transitionContext)
        maskLayerAnimation.delegate = self
        maskLayer.addAnimation(maskLayerAnimation, forKey: "path")
    }

}

override func animationDidStop(anim: CAAnimation!, finished flag: Bool) {
    self.transitionContext?.completeTransition(!self.transitionContext!.transitionWasCancelled())
    self.transitionContext?.viewControllerForKey(UITransitionContextFromViewControllerKey)?.view.layer.mask = nil
}

```

使用

```

var transDelegate : ModelTransitionDelegate = ModelTransitionDelegate()

var helpNav = UINavigationController(rootViewController: tipsVC)
helpNav.transitioningDelegate = self.transDelegate

self.presentViewController(helpNav, animated: true, completion:
{ () -> Void in

})

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-31 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

地图 - MKMapView 地图开发相关总结

## 关键字

MKMapView \ CLLocationCoordinate2D \ CLLocation

## 需求场景

1. 需要实现地图显示需求时

## 参考链接

(无)

## 详细内容

### 1. 坐标

1. *CLLocationCoordinate2D* 与 *NSString* 之间的相互转换 ```objectiveC`
2. `(CLLocationCoordinate2D) get2DCoordFromString:(NSString)coordString {  
 CLLocationCoordinate2D location; NSArray coordArray = [coordString  
 componentsSeparatedByString: @","]; location.latitude = ((NSNumber  
 )coordArray[0]).doubleValue; location.longitude = ((NSNumber  
 )coordArray[1]).doubleValue;  
  
 return location; }`

```
3. (NSString ) getStringFrom2DCoord :(CLLocationCoordinate2D)location {  
    NSString aString = [NSString  
        stringWithFormat:@"%f,%f",location.latitude,location.longitude]; return  
    aString; } ````
```

## 2. 大头针

```
1. 定义一个遵循 MKAnnotation 协议的大头针数据模型 ````swift // //  
REXAnnotation.swift // REX // // Created by Alfred Jiang on 3/27/15. //  
Copyright (c) 2015 REX. All rights reserved. //
```

```
import UIKit import MapKit  
  
class REXAnnotation: NSObject,MKAnnotation {  
  
    var coordinate: CLLocationCoordinate2D  
    var markerData: NSDictionary  
    //自定义一个初始化方法  
    init(coordinate: CLLocationCoordinate2D, markerData: NSDictionary)  
    {  
        self.markerData = markerData  
        self.coordinate = coordinate  
    }  
}
```

## 2. 添加到地图上

```
```swift
//定义地图
var mapViewLocation: MKMapView!

//添加大头针
var aLocationObject : REXAnnotation = REXAnnotation(coordinate:
centerLocation, markerData : NSDictionary())
self.mapViewLocation.addAnnotation(aLocationObject)

//实现显示方法
//MARK: - MapKit

func mapView(mapView: MKMapView!, didAddAnnotationViews views: [
AnyObject]!) {
    for mkaview in views
    {
        if mkaview is MKPinAnnotationView
        {
            (mkaview as MKPinAnnotationView).pinColor = MKPinAnn
otationColor.Red
        }
    }
}
```

## 3. 地图以某坐标居中显示

1. MKMapView+MapViewUtil.h  
```objectivec // // MKMapView+MapViewUtil.h //  
REX // // Created by Alfred Jiang on 3/27/15. // Copyright (c) 2015 REX. All  
rights reserved. //

# import

@interface MKMapView (MapViewUtil)

- (void)setCenterCoordinate:(CLLocationCoordinate2D)centerCoordinate

```
    zoomLevel:(NSUInteger)zoomLevel  
    animated:(BOOL)animated;
```

```
@end
```

```
2. MKMapView+MapViewUtil.m  
```objective-c  
//  
// MKMapView+MapViewUtil.m  
// REX  
//  
// Created by Alfred Jiang on 3/27/15.  
// Copyright (c) 2015 REX. All rights reserved.  
  
#import "MKMapView+MapViewUtil.h"  
  
#define MERCATOR_OFFSET 268435456  
#define MERCATOR_RADIUS 85445659.44705395  
  
@implementation MKMapView (MapViewUtil)  
  
#pragma mark -  
#pragma mark Map conversion methods  
  
- (double)longitudeToPixelSpaceX:(double)longitude  
{  
    return round(MERCATOR_OFFSET + MERCATOR_RADIUS * longitude *  
M_PI / 180.0);  
}  
  
- (double)latitudeToPixelSpaceY:(double)latitude  
{  
    return round(MERCATOR_OFFSET - MERCATOR_RADIUS * logf((1 + s  
inf(latitude * M_PI / 180.0)) / (1 - sinf(latitude * M_PI / 180.  
0))) / 2.0);  
}
```

```

- (double)pixelSpaceXToLongitude:(double)pixelX
{
    return ((round(pixelX) - MERCATOR_OFFSET) / MERCATOR_RADIUS)
    * 180.0 / M_PI;
}

- (double)pixelSpaceYToLatitude:(double)pixelY
{
    return (M_PI / 2.0 - 2.0 * atan(exp((round(pixelY) - MERCATOR_OFFSET) / MERCATOR_RADIUS))) * 180.0 / M_PI;
}

#pragma mark -
#pragma mark Helper methods

- (MKCoordinateSpan)coordinateSpanWithMapView:(MKMapView *)mapView
                                         centerCoordinate:(CLLocationCoordinate2D)centerCoordinate
                                         andZoomLevel:(NSUInteger)zoomLevel
{
    // convert center coordinate to pixel space
    double centerPixelX = [self longitudeToPixelSpaceX:centerCoordinate.longitude];
    double centerPixelY = [self latitudeToPixelSpaceY:centerCoordinate.latitude];

    // determine the scale value from the zoom level
    NSInteger zoomExponent = 20 - zoomLevel;
    double zoomScale = pow(2, zoomExponent);

    // scale the map's size in pixel space
    CGSize mapSizeInPixels = mapView.bounds.size;
    double scaledMapWidth = mapSizeInPixels.width * zoomScale;
    double scaledMapHeight = mapSizeInPixels.height * zoomScale;

    // figure out the position of the top-left pixel
    double topLeftPixelX = centerPixelX - (scaledMapWidth / 2);
    double topLeftPixelY = centerPixelY - (scaledMapHeight / 2);
}

```

```

    // find delta between left and right longitudes
    CLLocationDegrees minLng = [self pixelSpaceXToLongitude:topLeftPixelX];
    CLLocationDegrees maxLng = [self pixelSpaceXToLongitude:topLeftPixelX + scaledMapWidth];
    CLLocationDegrees longitudeDelta = maxLng - minLng;

    // find delta between top and bottom latitudes
    CLLocationDegrees minLat = [self pixelSpaceYToLatitude:topLeftPixelY];
    CLLocationDegrees maxLat = [self pixelSpaceYToLatitude:topLeftPixelY + scaledMapHeight];
    CLLocationDegrees latitudeDelta = -1 * (maxLat - minLat);

    // create and return the lat/lng span
    MKCoordinateSpan span = MKCoordinateSpanMake(latitudeDelta,
longitudeDelta);
    return span;
}

#pragma mark -
#pragma mark Public methods

- (void)setCenterCoordinate:(CLLocationCoordinate2D)centerCoordinate
    zoomLevel:(NSUInteger)zoomLevel
    animated:(BOOL)animated
{
    // clamp large numbers to 28
    zoomLevel = MIN(zoomLevel, 28);

    // use the zoom level to compute the region
    MKCoordinateSpan span = [self coordinateSpanWithMapView:self
centerCoordinate:centerCoordinate andZoomLevel:zoomLevel];
    MKCoordinateRegion region = MKCoordinateRegionMake(centerCoordinate, span);

    // set the region like normal
    [self setRegion:region animated:animated];
}

```

```
}
```

```
@end
```

## 1. 使用

```
self.mapViewLocation.setCenterCoordinate(centerLocation, zoom  
Level: 14, animated: true)
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-23 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

地图 - 为 Xcode 工程添加自定义初始化位置（载入 gpx 文件）

## 关键字

地图 \ 坐标 \ 初始位置 \ gpx

## 需求场景

1. 在模拟器中实现地图位置模拟
2. 在示例场景中实现位置模拟
3. 地图读取默认位置需求

## 参考链接

1. [XCode 4.2 地点模拟技巧](#)

## 详细内容

1. 创建 **gpx** 文件，可命名为 "**TestLocation.gpx**"

```
New -> File -> iOS -> Resource -> GPX File
```

2. 添加坐标

```
<?xml version="1.0"?>

<gpx version="1.1" creator="Xcode">

<wpt lat="37.331705" lon="-122.030237">
    <name>Cupertino</name>
</wpt>

</gpx>
```

3. 启动模拟器，可点击下方位置出现 "**TestLocation**" 选项

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注       |
|----|------------|--------------|----------|
| 1  | 2016-02-29 | Alfred Jiang | -        |
| 1  | 2016-03-08 | Alfred Jiang | 更新 OC 调用 |

## 方案名称

地图 - 使用 Eviltransform 进行火星坐标转换（大陆地区地理坐标偏移问题）

## 关键字

地图 \ Eviltransform \ 坐标 \ 偏移

## 需求场景

1. 需要解决中国大陆地图坐标偏移问题

## 参考链接

1. [Why You Can't Trust GPS in China](#)
2. [GitHub - Eviltransform](#)

## 详细内容

**Swift Version :** 更新版，支持 OC 调用

```
//  
// LocationTransform.swift  
// PSWalker  
//  
// Created by viktyz on 16/3/8.  
// Copyright © 2016年 Alfred Jiang. All rights reserved.  
//
```

```

import Foundation

@objc public class LocationTransform : NSObject {

    static let π = M_PI, latKey = "lat", lonKey = "lon"

    static func isOutOfChina(lat lat: Double, lon: Double) -> Bool {
        if lon < 72.004 || lon > 137.8347 {
            return true
        }
        if lat < 0.8293 || lat > 55.8271 {
            return true
        }
        return false
    }

    static func transformLat(x x: Double, y: Double) -> Double {
        var ret = -100.0 + 2.0 * x + 3.0 * y + 0.2 * y * y
        ret += 0.1 * x * y + 0.2 * sqrt(abs(x))
        ret += (20.0 * sin(6.0 * x * π) + 20.0 * sin(2.0 * x * π
)) * 2.0 / 3.0
        ret += (20.0 * sin(y * π) + 40.0 * sin(y / 3.0 * π)) * 2
.0 / 3.0
        ret += (160.0 * sin(y / 12.0 * π) + 320 * sin(y * π / 30
.0)) * 2.0 / 3.0
        return ret
    }

    static func transformLon(x x: Double, y: Double) -> Double {
        var ret = 300.0 + x + 2.0 * y + 0.1 * x * x
        ret += 0.1 * x * y + 0.1 * sqrt(abs(x))
        ret += (20.0 * sin(6.0 * x * π) + 20.0 * sin(2.0 * x * π
)) * 2.0 / 3.0
        ret += (20.0 * sin(x * π) + 40.0 * sin(x / 3.0 * π)) * 2
.0 / 3.0
        ret += (150.0 * sin(x / 12.0 * π) + 300.0 * sin(x / 30.0
* π)) * 2.0 / 3.0
        return ret
    }
}

```

```

    static func delta(lat: Double, lon: Double) -> (Double, Double) {
        let r = 6378245.0
        let ee = 0.00669342162296594323
        let radLat = lat / 180.0 * π
        var magic = sin(radLat)
        magic = 1 - ee * magic * magic
        let sqrtMagic = sqrt(magic)
        var dLat = transformLat(x: lon - 105.0, y: lat - 35.0)
        var dLon = transformLon(x: lon - 105.0, y: lat - 35.0)
        dLat = (dLat * 180.0) / ((r * (1 - ee)) / (magic * sqrtMagic) * π)
        dLon = (dLon * 180.0) / (r / sqrtMagic * cos(radLat) * π)
    }
    return (dLat, dLon)
}

```

//GCJ-02坐标用在谷歌地图，高德地图等中国地图服务。（百度地图要在GCJ-02基础上再加转换）

```

//输入WGS-84地球坐标(wgsLat, wgsLng)，转换为GCJ-02火星坐标(gcjLat, gcjLng)
static func wgs2gcj(wgsLat: Double, wgsLon: Double) -> [String: Double] {
    if isOutOfChina(lat: wgsLat, lon: wgsLon) {
        return [latKey: wgsLat, lonKey: wgsLon]
    }
    let (dLat, dLon) = delta(lat: wgsLat, lon: wgsLon)
    return [latKey: wgsLat + dLat, lonKey: wgsLon + dLon]
}

```

//输入GCJ-02火星坐标(gcjLat, gcjLng)，转换为WGS-84地球坐标(wgsLat, wgsLng)，输出的WGS-84坐标精度为1米到2米之间

```

static func gcj2wgs(gcjLat: Double, gcjLon: Double) -> [String: Double] {
    if isOutOfChina(lat: gcjLat, lon: gcjLon) {
        return [latKey: gcjLat, lonKey: gcjLon]
    }
    let (dLat, dLon) = delta(lat: gcjLat, lon: gcjLon)

```

```

        return [latKey: gcjLat - dLat, lonKey: gcjLon - dLon]
    }

    //输入GCJ-02火星坐标(gcjLat, gcjLng)，转换为WGS-84地球坐标(wgsLat, wgsLng)，输出的WGS-84坐标精度0.5米内
    static func gcj2wgs_exact(gcjLat: Double, gcjLon: Double) -> [String: Double] {
        let initDelta = 0.01, threshold = 0.000001
        var dLat = initDelta
        var dLon = initDelta
        var mLat = gcjLat - dLat
        var mLon = gcjLon - dLon
        var pLat = gcjLat + dLat
        var pLon = gcjLon + dLon
        var wgsLat = gcjLat, wgsLon = gcjLon
        for (var i = 0; i < 30; i++) {
            wgsLat = (mLat + pLat) / 2
            wgsLon = (mLon + pLon) / 2
            var tmp = wgs2gcj(wgsLat, wgsLon: wgsLon) as [String : Double]
            dLat = tmp[latKey]! - gcjLat
            dLon = tmp[lonKey]! - gcjLon
            if (abs(dLat) < threshold) && (abs(dLon) < threshold)
        } {
            return [latKey: wgsLat, lonKey: wgsLon]
        }
        if dLat > 0 {
            pLat = wgsLat
        } else {
            mLat = wgsLat
        }
        if dLon > 0 {
            pLon = wgsLon
        } else {
            mLon = wgsLon
        }
    }
    return [latKey: wgsLat, lonKey: wgsLon]
}

```

```

//GCJ-02坐标 转为 百度地图坐标
static func gcj2bd(gcjLat: Double, gcjLon: Double) -> [String: Double] {
    if isOutOfChina(lat: gcjLat, lon: gcjLon) {
        return [latKey: gcjLat, lonKey: gcjLon]
    }
    let x = gcjLon, y = gcjLat
    let z = sqrt(x * x + y * y) + 0.00002 * sin(y * π)
    let theta = atan2(y, x) + 0.000003 * cos(x * π)
    let bdLon = z * cos(theta) + 0.0065
    let bdLat = z * sin(theta) + 0.006
    return [latKey: bdLat, lonKey: bdLon]
}

//百度地图坐标 转为 GCJ-02坐标
static func bd2gcj(bdLat: Double, bdLon: Double) -> [String: Double] {
    if isOutOfChina(lat: bdLat, lon: bdLon) {
        return [latKey: bdLat, lonKey: bdLon]
    }
    let x = bdLon - 0.0065, y = bdLat - 0.006
    let z = sqrt(x * x + y * y) - 0.00002 * sin(y * π)
    let theta = atan2(y, x) - 0.000003 * cos(x * π)
    let gcjLon = z * cos(theta)
    let gcjLat = z * sin(theta)
    return [latKey: gcjLat, lonKey: gcjLon]
}

//WGS-84坐标 转为 百度地图坐标
static func wgs2bd(wgsLat: Double, wgsLon: Double) -> [String: Double] {
    let gcj = wgs2gcj(wgsLat, wgsLon) as [String: Double]
    return gcj2bd(gcj[latKey]!, gcjLon: gcj[lonKey]!)
}

//百度地图坐标 转为 WGS-84坐标
static func bd2wgs(bdLat: Double, bdLon: Double) -> [String: Double] {
    let gcj = bd2gcj(bdLat, bdLon: bdLon) as [String: Double]

```

```

    ]
        return gcj2wgs(gcj[latKey]!, gcjLon: gcj[lonKey]!)
    }

}

```

## Python Version

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-

import math


__all__ = ['wgs2gcj', 'gcj2wgs', 'gcj2wgs_exact',
           'distance', 'gcj2bd', 'bd2gcj', 'wgs2bd', 'bd2wgs']


def outOfChina(lat, lng):
    return not (72.004 <= lng <= 137.8347 and 0.8293 <= lat <= 5
5.8271)

def transformLat(x, y):
    ret = (-100.0 + 2.0 * x + 3.0 * y + 0.2 * y *
           y + 0.1 * x * y + 0.2 * math.sqrt(abs(x)))
    ret += (20.0 * math.sin(6.0 * x * math.pi) + 20.0 *
            math.sin(2.0 * x * math.pi)) * 2.0 / 3.0
    ret += (20.0 * math.sin(y * math.pi) + 40.0 *
            math.sin(y / 3.0 * math.pi)) * 2.0 / 3.0
    ret += (160.0 * math.sin(y / 12.0 * math.pi) + 320.0 *
            math.sin(y * math.pi / 30.0)) * 2.0 / 3.0
    return ret

def transformLon(x, y):
    ret = (300.0 + x + 2.0 * y + 0.1 * x * x +
           0.1 * x * y + 0.1 * math.sqrt(abs(x)))
    ret += (20.0 * math.sin(6.0 * x * math.pi) + 20.0 *

```

```

        math.sin(2.0 * x * math.pi)) * 2.0 / 3.0
ret += (20.0 * math.sin(x * math.pi) + 40.0 *
        math.sin(x / 3.0 * math.pi)) * 2.0 / 3.0
ret += (150.0 * math.sin(x / 12.0 * math.pi) + 300.0 *
        math.sin(x / 30.0 * math.pi)) * 2.0 / 3.0
return ret
}

def delta(lat, lng):
    a = 6378245.0
    ee = 0.00669342162296594323
    dLat = transformLat(lng - 105.0, lat - 35.0)
    dLng = transformLon(lng - 105.0, lat - 35.0)
    radLat = lat / 180.0 * math.pi
    magic = math.sin(radLat)
    magic = 1 - ee * magic * magic
    sqrtMagic = math.sqrt(magic)
    dLat = (dLat * 180.0) / ((a * (1 - ee)) / (magic * sqrtMagic
) * math.pi)
    dLng = (dLng * 180.0) / (a / sqrtMagic * math.cos(radLat) *
math.pi)
    return dLat, dLng

def wgs2gcj(wgsLat, wgsLng):
    if outOfChina(wgsLat, wgsLng):
        return wgsLat, wgsLng
    else:
        dlat, dlng = delta(wgsLat, wgsLng)
        return wgsLat + dlat, wgsLng + dlng

def gcj2wgs(gcjLat, gcjLng):
    if outOfChina(gcjLat, gcjLng):
        return gcjLat, gcjLng
    else:
        dlat, dlng = delta(gcjLat, gcjLng)
        return gcjLat - dlat, gcjLng - dlng
}

```

```

def gcj2wgs_exact(gcjLat, gcjLng):
    initDelta = 0.01
    threshold = 0.000001
    dLat = dLng = initDelta
    mLat = gcjLat - dLat
    mLng = gcjLng - dLng
    pLat = gcjLat + dLat
    pLng = gcjLng + dLng
    for i in range(30):
        wgsLat = (mLat + pLat) / 2
        wgsLng = (mLng + pLng) / 2
        tmplat, tmplng = wgs2gcj(wgsLat, wgsLng)
        dLat = tmplat - gcjLat
        dLng = tmplng - gcjLng
        if abs(dLat) < threshold and abs(dLng) < threshold:
            return wgsLat, wgsLng
        if dLat > 0:
            pLat = wgsLat
        else:
            mLat = wgsLat
        if dLng > 0:
            pLng = wgsLng
        else:
            mLng = wgsLng
    return wgsLat, wgsLng

def distance(latA, lngA, latB, lngB):
    earthR = 6371000
    x = (math.cos(latA * math.pi / 180) * math.cos(latB * math.pi / 180) *
          math.cos((lngA - lngB) * math.pi / 180))
    y = math.sin(latA * math.pi / 180) * math.sin(latB * math.pi / 180)
    s = x + y
    if s > 1:
        s = 1
    if s < -1:
        s = -1
    alpha = math.acos(s)

```

```

distance = alpha * earthR
return distance

def gcj2bd(gcjLat, gcjLng):
    if outOfChina(gcjLat, gcjLng):
        return gcjLat, gcjLng

    x = gcjLng
    y = gcjLat
    z = math.hypot(x, y) + 0.00002 * math.sin(y * math.pi)
    theta = math.atan2(y, x) + 0.000003 * math.cos(x * math.pi)
    bdLng = z * math.cos(theta) + 0.0065
    bdLat = z * math.sin(theta) + 0.006
    return bdLat, bdLng

def bd2gcj(bdLat, bdLng):
    if outOfChina(bdLat, bdLng):
        return bdLat, bdLng

    x = bdLng - 0.0065
    y = bdLat - 0.006
    z = math.hypot(x, y) - 0.00002 * math.sin(y * math.pi)
    theta = math.atan2(y, x) - 0.000003 * math.cos(x * math.pi)
    gcjLng = z * math.cos(theta)
    gcjLat = z * math.sin(theta)
    return gcjLat, gcjLng

def wgs2bd(wgsLat, wgsLng):
    return gcj2bd(*wgs2gcj(wgsLat, wgsLng))

def bd2wgs(bdLat, bdLng):
    return gcj2wgs(*bd2gcj(bdLat, bdLng))

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-06-12 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

字体 - 添加自定义字体

## 关键字

字体 \ 自定义字体 \ 非系统字体

## 需求场景

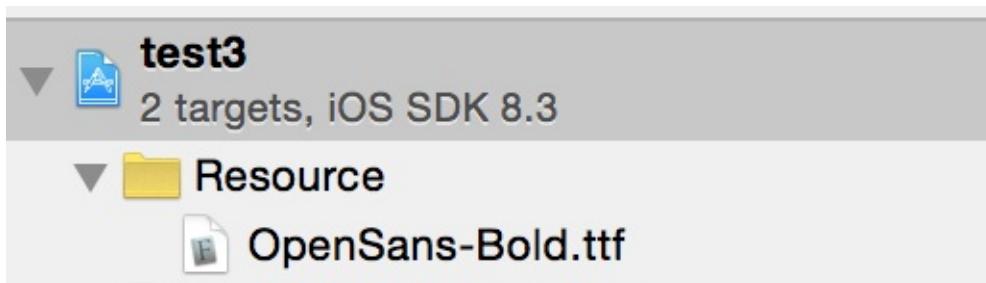
1. 需要为 App 提供系统之外的字体时

## 参考链接

1. CSDN - ios开发 添加自定义字体
2. 字体下载网站一

## 详细内容

1. 下载需要的 .ttf 格式字体，并添加入工程



2. 在 .plist 中添加如下字段

| ▼ Fonts provided by application | Array  | (1 item)          |
|---------------------------------|--------|-------------------|
| Item 0                          | String | OpenSans-Bold.ttf |

```
<key>UIAppFonts</key>
<array>
<string>OpenSans-Bold.ttf</string>
</array>
```

3. 确保 *Project -> Targets -> Build Phases -> Copy Bundle Resources* 中字体已加入



4. 在工程中执行下面的代码找到字体对应名称

```
NSMutableArray *familyNames = [[NSMutableArray alloc] initWithArray:[UIFont
familyNames]];
NSMutableArray *fontNames;
NSInteger indFamily, indFont;
for (indFamily=0; indFamily<[familyNames count]; ++indFamily)
{
    NSLog(@"%@", [familyNames objectAtIndex:indFamily]);
    fontNames = [[NSMutableArray alloc] initWithArray:
        [UIFont fontNamesForFamilyName:
            [familyNames objectAtIndex:indFamily]]];
    for (indFont=0; indFont<[fontNames count]; ++indFont)
    {
        NSLog(@"%@", [fontNames objectAtIndex:indFont]);
    }
}
```



5. 现在就可以像调用系统字体一样的调用添加字体了

```
aLabel.font= [UIFont fontWithName:@"OpenSans-Bold" size:14];
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-05-04 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

工具 - Markdown 简明教程

## 关键字

工具 \ Markdown \ 教程

## 需求场景

1. 学习 Markdown 快速入门教程

## 参考链接

(见详细内容)

## 详细内容

1. [《CSDN Markdown简明教程》课程上线！](#)
2. [CSDN Markdown简明教程1-关于Markdown](#)
3. [CSDN Markdown简明教程2-基本使用](#)
4. [CSDN Markdown简明教程3-表格和公式](#)
5. [CSDN Markdown简明教程4-UML图](#)
6. [CSDN Markdown简明教程5-快速上手](#)

本收录转载自 CSDN 王海庆的 Markdown 学习教程，为表示对原作者的尊重，保留以下声明

## 声明

前端开发whqet，关注前端开发，分享相关资源。csdn专家博客，王海庆希望能对您有所帮助，限于作者水平有限，出错难免，欢迎拍砖！欢迎任何形式的转载，烦请注明装载，保留本段文字。

本文原文链接，<http://blog.csdn.net/whqet/article/details/44318301>

欢迎大家访问独立博客<http://whqet.github.io>

## 效果图

(无)

## 备注

- GitHub - SublimeText-Markdown/MarkdownEditing:SublimeText 的 Markdown 插件

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-25 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

工具 - 使用 Cuttermen 插件进行快速切图

## 关键字

工具 \ Cuttermen \ Photoshop \ 切图 \ 插件

## 需求场景

1. 需要 PSD 文件自己切图时

## 参考链接

1. [Cuttermen 官网](#)

## 详细内容

以 Adobe Photoshop CC 2014 为例

1. 进入 **Cuttermen** 网站 [下载](#) 对应系统和版本的 **Cuttermen** 插件
2. [安装 Adobe Extension Manager](#)
3. [升级 Adobe Extension Manager 至 7.2+](#)
4. 通过 **Adobe Extension Manager** 安装 **Cuttermen** 插件
5. [注册 Cuttermen 账号 并使用插件](#)

**6. 查看 [更多帮助](#)**

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-10-18 | Alfred Jiang | -  |

## 方案名称

工具 - 使用 iBackupbot 查看非越狱真机应用文件

## 关键字

工具 \ iBackupbot \ 查看文件 \ 真机目录

## 需求场景

1. 解决 iOS 8 之后 PP 助手、iTools 等工具难以查看非越狱机器 App 目录问题
2. 修改非越狱机器下 App 中文件

## 参考链接

1. [BackupBot](#)(推荐)

## 详细内容

iOS 8 之后大部分应用难以查看和修改非越狱机器下 App 中文件，iBackupbot 可以通过曲线方式解决这一问题

使用方法：

1. 首先需要确保机器安装 [iTunes](#) 和 [iBackupbot](#)
2. 连接真机并通过 [iTunes](#) 备份真机所有数据（主要是确保 App 数据备份至电脑）
3. 通过 [iBackupbot](#) 载入电脑上的备份信息，导出并操作
4. 修改完毕保存回去，通过 [iTunes](#) 恢复至真机

整个流程如果涉及到修改保存确实比较繁琐，但确实能满足查看需求，算是 iOS 8 之后 iTools 、 PP 助手等工具比较好的替代方案。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-15 | Alfred Jiang | -  |

## 方案名称

工具 - 使用 `objc_dep` 检查项目中的导入依赖(Import Dependencies)

## 关键字

工具 \ `objc_dep` \ 导入依赖 \ Import Dependencies

## 需求场景

1. 重构工程，实现代码的疏耦合

## 参考链接

1. Segmentfault - Objective-C 项目重构利器：把项目中的导入依赖（Import Dependencies）图示化
2. GitHub - `objc_dep`
3. GitHub - ObjectGraph-Xcode

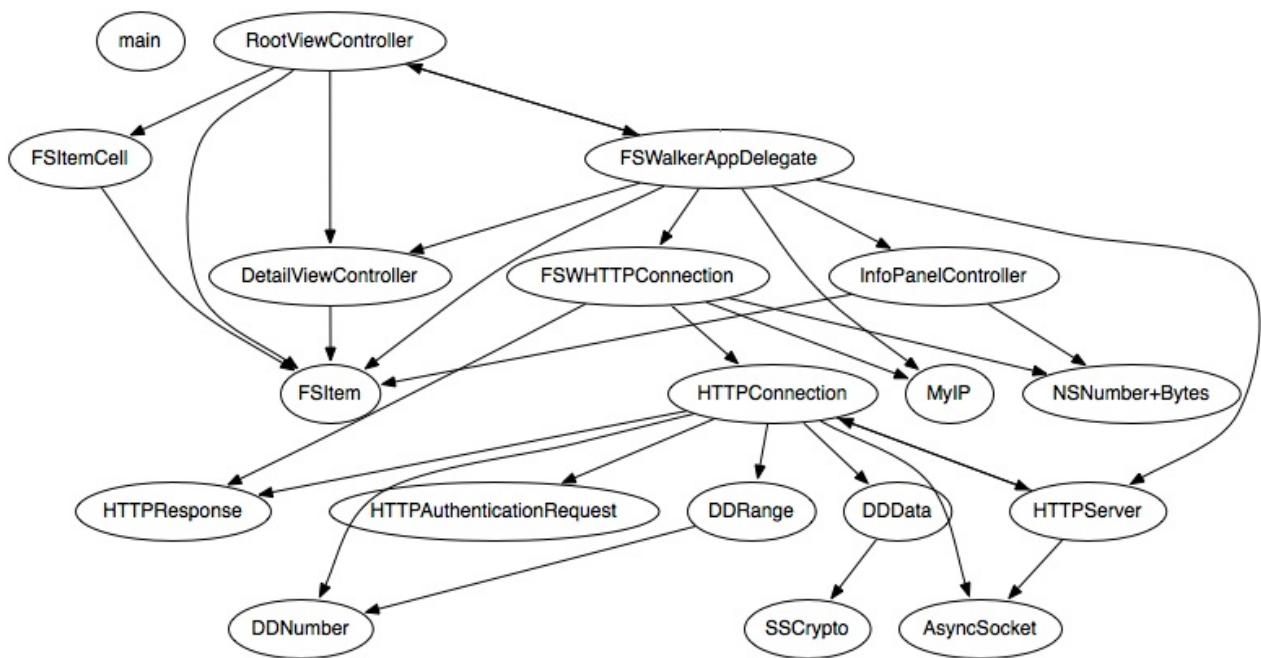
## 详细内容

1. 从 [GitHub - `objc\_dep`](#) 获取最新的 `objc_dep.py`

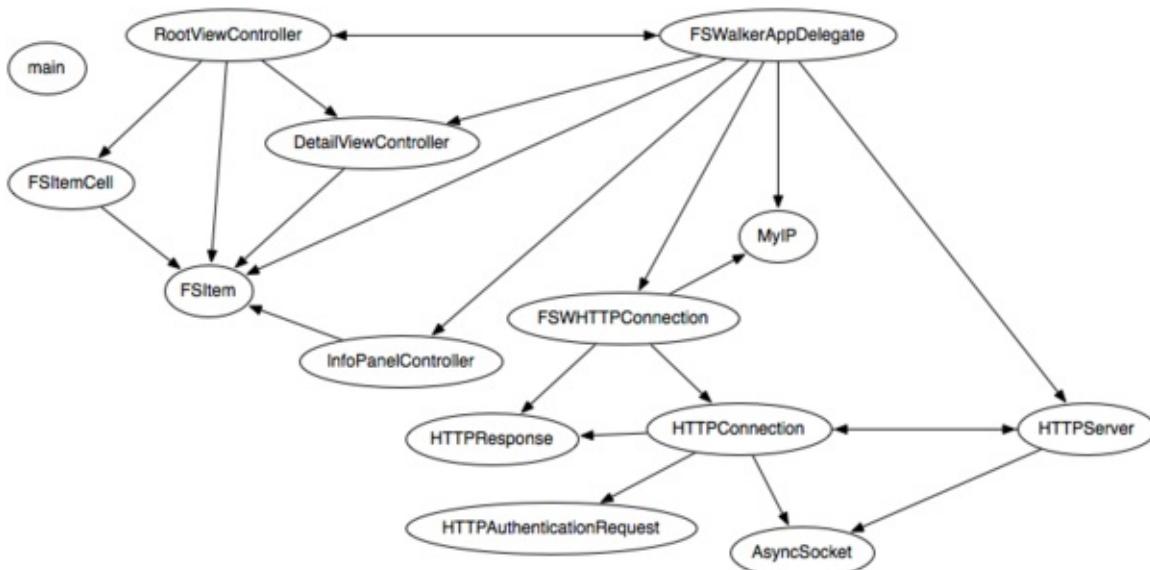
2. 执行以下命令生成导入依赖关系图

```
$ python objc_dep.py /path/to/YourProject > relation.dot
```

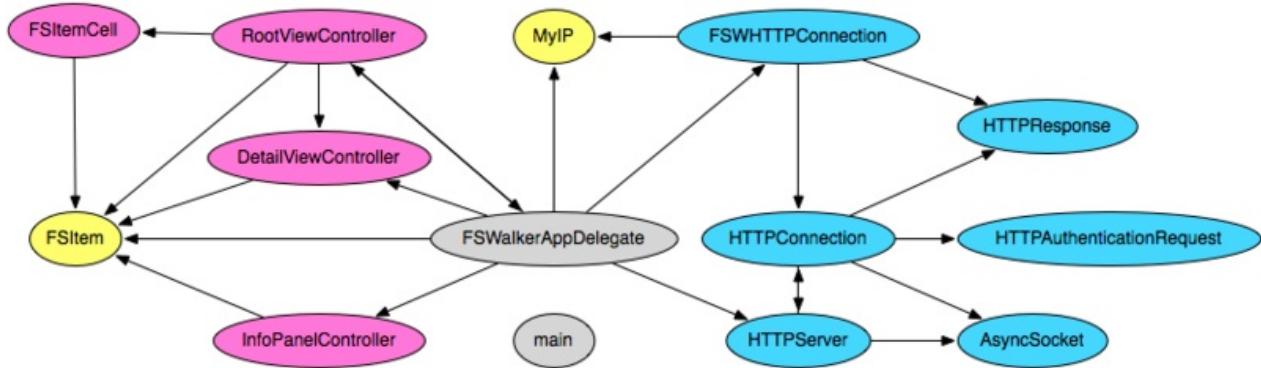
3. 在 [OmniGraffle](#) 中打开 `relation.dot`



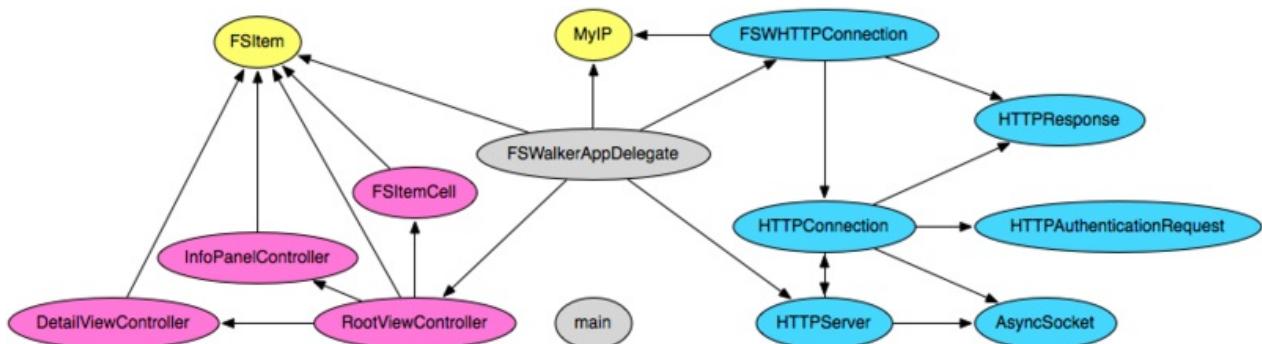
#### 4. 移除 categories



#### 5. 将关联类分组



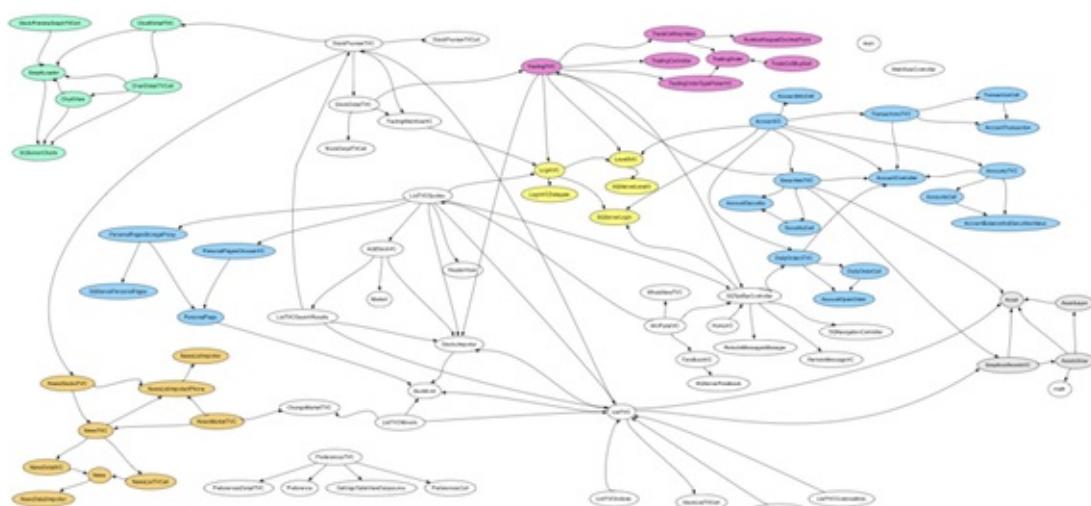
#### 6. 检查有问题的依赖



7. 额外选项，下面会包括所有以 "**Internal**" 或 "**secret**" 开头的文件，同时所有 **subdir1** 和 **subdir2** 目录下的文件会被忽略

```
$ python objc_dep.py /path/to/repo -x "^(Internal|secret)" -i subdir1 subdir2 > graph.dot
```

## 效果图



## 备注

```
#!/usr/bin/python

# Nicolas Seriot
# 2011-01-06 -> 2011-12-16
# https://github.com/nst/objc_dep/

"""

Input: path of an Objective-C project
Output: import dependencies Graphviz format

```

```

Typical usage: $ python objc_dep.py /path/to/project [-x regex]
[-i subfolder [subfolder ...]] > graph.dot
The .dot file can be opened with Graphviz or OmniGraffle.
- red arrows: .pch imports
- blue arrows: two ways imports
"""

import sys
import os
from sets import Set
import re
from os.path import basename
import argparse

local_regex_import = re.compile("^\s*#(?:import|include)\s+\"(?P<filename>\S*)(?P<extension>\.(?:h|hpp|hh))?\\"")
system_regex_import = re.compile("^\s*#(?:import|include)\s+[\\"<](?P<filename>\S*)(?P<extension>\.(?:h|hpp|hh))?[\"\>]\"")

def gen_filenames_imported_in_file(path, regex_exclude, system,
extensions):
    for line in open(path):
        results = re.search(system_regex_import, line) if system
        else re.search(local_regex_import, line)
        if results:
            filename = results.group('filename')
            extension = results.group('extension') if results.gr
            oup('extension') else ""
            if regex_exclude is not None and regex_exclude.searc
            h(filename + extension):
                continue
            yield (filename + extension) if extension else filen
            ame

def dependencies_in_project(path, ext, exclude, ignore, system,
extensions):
    d = {}

    regex_exclude = None
    if exclude:

```

```
    regex_exclude = re.compile(exclude)

    for root, dirs, files in os.walk(path):

        if ignore:
            for subfolder in ignore:
                if subfolder in dirs:
                    dirs.remove(subfolder)

    objc_files = (f for f in files if f.endswith(ext))

    for f in objc_files:

        filename = f if extensions else os.path.splitext(f)[0]
        if regex_exclude is not None and regex_exclude.search(filename):
            continue

        if filename not in d:
            d[filename] = Set()

        path = os.path.join(root, f)

        for imported_filename in gen_filenames_imported_in_file(path, regex_exclude, system, extensions):
            if imported_filename != filename and '+' not in imported_filename and '+' not in filename:
                imported_filename = imported_filename if extensions else os.path.splitext(imported_filename)[0]
                d[filename].add(imported_filename)

    return d

def dependencies_in_project_with_file_extensions(path, exts, exclude, ignore, system, extensions):
    d = {}

    for ext in exts:
```

```
d2 = dependencies_in_project(path, ext, exclude, ignore,
system, extensions)
    for (k, v) in d2.iteritems():
        if not k in d:
            d[k] = Set()
            d[k] = d[k].union(v)

return d

def two_ways_dependencies(d):
    two_ways = Set()

    # d is {'a1':[b1, b2], 'a2':[b1, b3, b4], ...}

    for a, l in d.iteritems():
        for b in l:
            if b in d and a in d[b]:
                if (a, b) in two_ways or (b, a) in two_ways:
                    continue
                if a != b:
                    two_ways.add((a, b))

    return two_ways

def untraversed_files(d):
    dead_ends = Set()

    for file_a, file_a_dependencies in d.iteritems():
        for file_b in file_a_dependencies:
            if not file_b in dead_ends and not file_b in d:
                dead_ends.add(file_b)

    return dead_ends

def category_files(d):
    d2 = {}
    l = []
```

```

for k, v in d.iteritems():
    if not v and '+' in k:
        l.append(k)
    else:
        d2[k] = v

return l, d2

def referenced_classes_from_dict(d):
    d2 = {}

    for k, deps in d.iteritems():
        for x in deps:
            d2.setdefault(x, Set())
            d2[x].add(k)

    return d2

def print_frequencies_chart(d):

    lengths = map(lambda x:len(x), d.itervalues())
    if not lengths: return
    max_length = max(lengths)

    for i in range(0, max_length+1):
        s = "%2d | %s\n" % (i, '*'*lengths.count(i))
        sys.stderr.write(s)

    sys.stderr.write("\n")

    l = [Set() for i in range(max_length+1)]
    for k, v in d.iteritems():
        l[len(v)].add(k)

    for i in range(0, max_length+1):
        s = "%2d | %s\n" % (i, ", ".join(sorted(list(l[i]))))
        sys.stderr.write(s)

def dependencies_in_dot_format(path, exclude, ignore, system, extensions):

```

```

d = dependencies_in_project_with_file_extensions(path, ['.h',
, '.hh', '.hpp', '.m', '.mm', '.c', '.cc', '.cpp'], exclude, ignore, system, extensions)

two_ways_set = two_ways_dependencies(d)
untraversed_set = untraversed_files(d)

category_list, d = category_files(d)

pch_set = dependencies_in_project(path, '.pch', exclude, ignore, system, extensions)

#
# number of imports\n\n)
print_frequencies_chart(d)

#\n# times the class is imported\n\n)
d2 = referenced_classes_from_dict(d)
print_frequencies_chart(d2)

#
l = []
l.append("digraph G {")
l.append("\tnode [shape=box];")

for k, deps in d.iteritems():
    if deps:
        deps.discard(k)

    if len(deps) == 0:
        l.append("\t\"%s\" -> {};" % (k))

    for k2 in deps:
        if not ((k, k2) in two_ways_set or (k2, k) in two_ways_set):
            l.append("\t\"%s\" -> \"%s\";" % (k, k2))

```

```

l.append("\t")
for (k, v) in pch_set.iteritems():
    l.append("\t\"%s\" [color=red];" % k)
    for x in v:
        l.append("\t\"%s\" -> \"%s\" [color=red];" % (k, x))

l.append("\t")
l.append("\tedge [color=blue, dir=both];")

for (k, k2) in two_ways_set:
    l.append("\t\"%s\" -> \"%s\";" % (k, k2))

for k in untraversed_set:
    l.append("\t\"%s\" [color=gray, style=dashed, fontcolor=gray]" % k)

if category_list:
    l.append("\t")
    l.append("\tedge [color=black];")
    l.append("\tnode [shape=plaintext];")
    l.append("\t\"Categories\" [label=\"%s\"];" % "\n".join(category_list))

if ignore:
    l.append("\t")
    l.append("\tnode [shape=box, color=blue];")
    l.append("\t\"Ignored\" [label=\"%s\"];" % "\n".join(ignore))

l.append("}\n")
return '\n'.join(l)

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("-x", "--exclude", nargs='?', default='',
    help="regular expression of substrings to exclude from module names")
    parser.add_argument("-i", "--ignore", nargs='*', help="list of subfolder names to ignore")
    parser.add_argument("-s", "--system", action='store_true', d

```

```
efault=False, help="include system dependencies")
    parser.add_argument("-e", "--extensions", action='store_true'
, default=False, help="print file extensions")
    parser.add_argument("project_path", help="path to folder hie
rarchy containing Objective-C files")
args= parser.parse_args()

print dependencies_in_dot_format(args.project_path, args.exc
lude, args.ignore, args.system, args.extensions)

if __name__=='__main__':
    main()
```

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-24 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

工具 - 使用 VisualJSON 进行 JSON 格式网络接口 API 测试

## 关键字

工具 \ Mac 系统 \ MAC OS \ VisualJSON \ 网络请求 \ API \ POST \ GET

## 需求场景

1. 需要进行返回数据为 JSON 类型的网络接口 API 测试时

## 参考链接

1. Appstore 下载地址
2. GitHub - VisualJSON

## 详细内容

### 1. 发送 **GET** 请求

1. 在 Address 中输入请求连接
2. 点击 Show Details 展开参数输入页
3. 在 File-Value 中输入 GET 请求参数 (亦可直接在 QueryData 中输入 GET 请求参数)

### 2. 发送 **POST** 请求

1. 在 Address 中输入请求连接

2. 点击 **Show Details** 展开参数输入页
3. 在 **RawText** 中输入 POST Json 格式请求参数

## 效果图

(无)

## 备注

另有 [VisualXML](#) 工具可以实现 XML 格式网络请求 API 测试

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-20 | Alfred Jiang | -  |

## 方案名称

工具 - 使用 Xcodebuild 进行命令行打包

## 关键字

工具 \ Xcodebuild \ 命令行打包 \ 自动化打包

## 需求场景

1. 为实现打包脚本做准备，简化打包流程

## 参考链接

1. 使用 Xcodebuild + Jenkins + Apache 做 iOS 持续集成
2. 动手搭建 iOS CI 环境之「了解 xcdebuild 命令」
3. 使用 xcdebuild 从 archive 导出 ipa(推荐)
4. GitHub - spritefly/ios-builder
5. 简书 - 敲一下enter键，完成iOS的打包工作

## 详细内容

以工程名为 TestBuildProject.xcodeproj 的工程举例如下

首先，打开终端，进入工程文件 *TestBuildProject.xcodeproj* 所在同级目录

```
$ cd yourPathTo/TestBuildProject
```

然后，使用 Xcodebuild 进行命令行打包，主要分为三个步骤

### 1. **clean** 工程

```
$ xcodebuild clean -project pathtoyour/TestBuildProject.xcodeproj -alltargets
```

命令解释：该行命令是对工程下全部 **targets** 进行 **clean** 操作

## 2. 生成 **.xcarchive** 文件，该文件包含用于解析 **crash** 日志的 **.dSYM** 文件和工程编译完成的 **.app** 文件

```
$ xcodebuild -project pathtoyour/TestBuildProject.xcodeproj -archivePath  
'~/Desktop/TestBuildProject.xcarchive' -sdk iphoneos -destination  
'generic/platform=iOS' -scheme "TestBuildProject" -configuration 'Release'  
archive
```

命令解释：该行命令是将当前目录下的 **TestBuildProject.xcodeproj** 工程编译生成 **TestBuildProject.xcarchive** 文件并输出到桌面，生成参数包括 **-sdk** 为 **iphoneos**, **-configuration** 为 **Release**

注意：

1. 若包含 **xcworkspace** 文件，务必添加 **-workspace TestBuildProject.xcworkspace** 参数
  2. 执行命令前务必确认打包证书与相关 **Provisioning Profile** 文件配置正确
- 3. 生成 .ipa 文件，该文件用于直接上传 **AppStore** 或者直接安装**

```
$ xcodebuild -exportArchive -exportFormat IPA -archivePath  
'~/Desktop/TestBuildProject.xcarchive' -exportPath  
'~/Desktop/TestBuildProject.ipa' -exportProvisioningProfile 'DistributionAll'
```

命令解释：该行命令是将桌面上的 **TestBuildProject.xcarchive** 文件编译输出为可供安装的 **TestBuildProject.ipa** 文件

## 3. [GitHub - spritefly/ios-builder](#)

`ios_build.config`

```
# used to name the ipa
APP_NAME=Demo

# directory to store exported ipa
APP_DIR=~/Desktop/app

# the absolute path to the project directory
PROJECT_DIR=/Users/reohou/Desktop/Demo
# project name, the same as the .xcodeproj file name
PROJECT_NAME="Demo"

# scheme name
SCHEME_NAME="Demo"

# profile name
PROFILE_NAME="adhoctag20140813"

# configuration for archive
CONFIGURATION="Release"

# login password, to unlock login keychain
LOGIN_PASSWORD="123456"
```

### ios\_build.sh

```
#!/bin/bash

function failed() {
    echo "Failed: $@" >&2
    exit 1
}

LOGIN_KEYCHAIN=~/Library/Keychains/login.keychain

script_dir_relative=`dirname $0`
script_dir=`cd ${script_dir_relative}; pwd`
echo "script_dir = ${script_dir}"

# read config
```

```
. ${script_dir}/ios_build.config

# unlock login keygen
security unlock-keychain -p ${LOGIN_PASSWORD} ${LOGIN_KEYCHAIN}
|| failed "unlock-keygen"

mkdir -pv ${APP_DIR} || failed "mkdir ${APP_DIR}"

cd ${PROJECT_DIR} || failed "cd ${PROJECT_DIR}"

rm -rf bin/*
mkdir -pv bin

# clean
xcodebuild clean -project ${PROJECT_NAME}.xcodeproj \
    -configuration ${CONFIGURATION} \
    -alltargets \
    || failed "xcodebuild clean"

# archive
xcodebuild archive -project ${PROJECT_NAME}.xcodeproj \
    -scheme ${SCHEME_NAME} \
    -destination generic/platform=iOS \
    -archivePath bin/${PROJECT_NAME}.xcarchive \
    || failed "xcodebuild archive"

# export ipa
xcodebuild -exportArchive -archivePath bin/${PROJECT_NAME}.xcarchive \
    -exportPath bin/${PROJECT_NAME} \
    -exportFormat ipa \
    -exportProvisioningProfile ${PROFILE_N
AME} \
    -verbose \
    || failed "xcodebuild export archive"

# move ipa to dest directory
timestamp=`date "+%Y%m%d%H"`

mv bin/${PROJECT_NAME}.ipa ${APP_DIR}/${APP_NAME}_${timestamp}.ipa || failed "mv ipa"
```

```
# clean bin files
echo "clean bin files ..."
rm -rf bin/*
echo "Done."
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-17 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

应用间通信 - App 之间共享数据的几种方法

## 关键字

应用间通信 \ 文档 \ App共享 \ 分享数据 \ 应用程序间通信

## 需求场景

1. 需要分享文档到另一个文件中打开时

## 参考链接

1. iOS app之间共享数据的几种方法
2. Enharmonic - Sharing Data Locally Between iOS Apps

## 详细内容

### 1. UIDocumentInteractionController

### 2. UIActivityViewController

参考一

参考二

### 3. Shared Keychain Access

这个要求app之间用的是同样的证书

#### 4. Custom URL Scheme

通过构造URL，把数据作为参数传递过去

参考一

5. Web Service 通过 dropbox 或者其他第三方的服务来共享数据

6. UIPasteboard + URL Scheme 通过 URL scheme 传递 UIPasteboard 的名称，然后通过 UIPasteboard 共享数据

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-10-08 | Alfred Jiang | -  |

## 方案名称

应用间通信 - URL Schemes

## 关键字

应用间通信 \ URL Schemes

## 需求场景

1. 在自己的应用中打开某个第三方应用或系统应用并调用指定功能

## 参考链接

1. 少数派 - URL Schemes 使用详解(推荐)
2. Stack Overflow - iOS Launching Settings -> Restrictions URL Scheme
3. MacRumors - iOS 8 Widget Customisations using Launcher
4. 开发者头条 - iOS 10 URL Scheme 列表

## 详细内容

关于 URL Schemes 的详细介绍可以[参考这里](#)

更多资料见参考链接

[iOS 10 URL Scheme 列表](#)

```
电池电量 Prefs:root=BATTERY_USAGE  
通用设置 Prefs:root=General  
存储空间 Prefs:root=General&path=STORAGE_ICLOUD_USAGE/DEVICE_STORAGE  
蜂窝数据 Prefs:root=MOBILE_DATA_SETTINGS_ID  
Wi-Fi 设置 Prefs:root=WIFI  
蓝牙设置 Prefs:root=Bluetooth  
定位设置 Prefs:root=Privacy&path=LOCATION  
辅助功能 Prefs:root=General&path=ACCESSIBILITY  
关于手机 Prefs:root=General&path=About  
键盘设置 Prefs:root=General&path=Keyboard  
显示设置 Prefs:root=DISPLAY  
声音设置 Prefs:root=Sounds  
App Store 设置 Prefs:root=STORE  
墙纸设置 Prefs:root=Wallpaper  
打开电话 Mobilephone://  
世界时钟 Clock-worldclock://  
闹钟 Clock-alarm://  
秒表 Clock-stopwatch://  
倒计时 Clock-timer://  
打开相册 Photos://
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-26 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

应用间通信 - 实现应用间互相调用与数据传递

## 关键字

应用间通信 \ 调用其他应用 \ 应用间调用

## 需求场景

1. 需要调用非系统应用时

## 参考链接

1. [CocoaChina - iOS开发长文--通讯录、蓝牙、内购、GameCenter、iCloud、Passbook系统服务开发汇总](#)

## 详细内容

假设你现在开发了一个应用 A，如果用户机器上已经安装了此应用，并且在应用 B 中希望能够直接打开 A。

1. 在 plist 文件中添加 URL types 节点并配置 URL Schemas 作为具体协议，配置 URL identifier 作为这个 URL 的唯一标识，确保应用 A 已经配置了 Url Types；

|                |            |                              |
|----------------|------------|------------------------------|
| ▼ URL types    | Array      | (1 item)                     |
| ▼ Item 0       | Dictionary | (2 items)                    |
| ▼ URL Schemes  | Array      | (1 item)                     |
| Item 0         | String     | cmj                          |
| URL identifier | String     | com.cmjstudio.iosapplication |

1. 使用 `openURL` 方法打开应用 B ``objectivec
2. `(IBAction)thirdPartyApplicationClick:(UIButton )sender { NSString url=@"cmj://myparams"; [self openUrl:url]; }`

```
-(void)openUrl:(NSString )urlStr{ //注意url中包含协议名称，iOS根据协议确定调用
哪个应用，例如发送邮件是“sms://”其中“//”可以省略写成“sms:”(其他协议也是如此)
NSURL url=[NSURL URLWithString:urlStr]; UIApplication *application=
[UIApplication sharedApplication]; if (![application canOpenURL:url]) { NSLog(@"无法打开%@",url); return; } [[UIApplication
sharedApplication] openURL:url]; }
```

3. 在应用 B 的 Appdelegate 中接受传递参数并打开

```
``objectivec
-(BOOL)application:(UIApplication *)application openURL:(NSURL *)
url sourceApplication:(NSString *)sourceApplication annotation:
(id)annotation{
    NSString *str=[NSString stringWithFormat:@"url:@"%@,source app
lication:@"%@,params:@"%@",url,sourceApplication,[url host]];
    NSLog(@"%@",str);
    return YES;//是否打开
}
```

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-25 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

应用间通信 - 文档导入导出实现

## 关键字

应用间通信 \ 文档 \ 文档导入 \ 文档导出 \ 其他应用共享打开

## 需求场景

1. 需要将自己应用内文档分享到其他应用打开时
2. 需要自己的应用打开其他应用中的文档时

## 参考链接

1. [Importing & Exporting Documents in iOS](#)
2. [CSDN - iOS App让自己的应用在其他应用中打开列表中显示](#)

## 详细内容

### 1. 导出自己应用内文档到其他应用打开

1. ViewController.h ```objective-c // // ViewController.h // test // // Created by Alfred Jiang on 4/25/15. // Copyright (c) 2015 Alfred Jiang. All rights reserved.  
//

## import

```
@interface ViewController : UIViewController
```

- (IBAction)btnDisplayFiles:(id)sender;
- (void)openDocumentIn;

```
@end
```

```
2. ViewController.m
```objective-c
//
//  ViewController.m
//  test
//
//  Created by Alfred Jiang on 4/25/15.
//  Copyright (c) 2015 Alfred Jiang. All rights reserved.
//


#import "ViewController.h"

@interface ViewController ()


@property (nonatomic, strong) UIDocumentInteractionController *documentController;
@end

@implementation ViewController
@synthesize documentController;

- (void)viewDidLoad {
    [super viewDidLoad];
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (IBAction)btnDisplayFiles:(id)sender
{
```

```

        [self openDocumentIn];
    }

-(void)openDocumentIn {
    NSString * filePath = [[NSBundle mainBundle] pathForResource:@"ee" ofType:@"pdf"];
    documentController = [UIDocumentInteractionController interactionControllerWithURL:[NSURL fileURLWithPath:filePath]];
    documentController.delegate = self;
    documentController.UTI = @"com.adobe.pdf";
    [documentController presentOpenInMenuFromRect:CGRectZero inView:self.view animated:YES];
}

-(void)documentInteractionController:(UIDocumentInteractionController *)controller willBeginSendingToApplication:(NSString *)application {
    NSLog(@"documentInteractionController : willBeginSendingToApplication");
}

-(void)documentInteractionController:(UIDocumentInteractionController *)controller didEndSendingToApplication:(NSString *)application {
    NSLog(@"documentInteractionController : didEndSendingToApplication");
}

-(void)documentInteractionControllerDidDismissOpenInMenu:(UIDocumentInteractionController *)controller {
    NSLog(@"documentInteractionControllerDidDismissOpenInMenu");
}

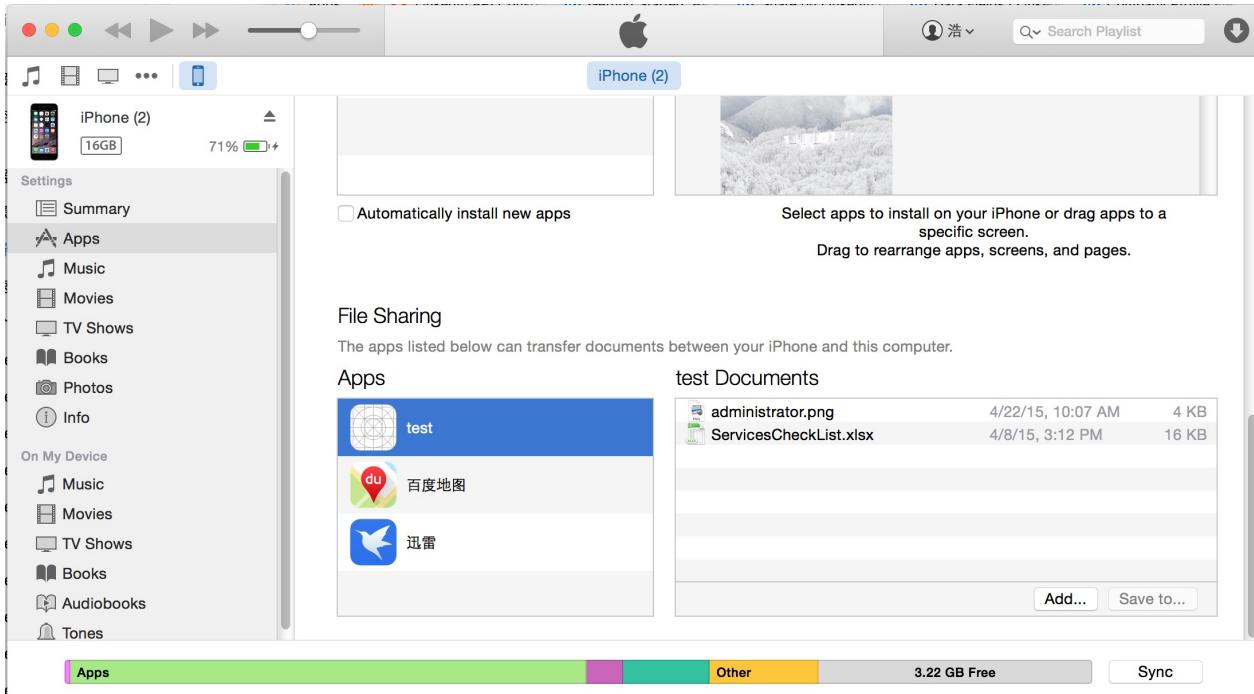
@end

```

## 2. 通过 iTunes 传输文档到手机并打开

- 在 info.plist 中增加 Application supports iTunes file sharing 为 YES (亦可设置 UIFileSharingEnabled 为 YES);

2. 链接 iPhone 至 iTunes，可在 iPhone -> Apps -> File Sharing 中看到自己应用；



1. 在 iTunes 选中自己应用，点击 Add... 按钮可添加文档至自己应用中；
2. 在自己应用中打开通过 iTunes 传输到应用中的文档 ``objective // // ViewController.h // test // // Created by Alfred Jiang on 4/25/15. // Copyright (c) 2015 Alfred Jiang. All rights reserved. //

## import

```

@interface ViewController : UIViewController

@property (weak, nonatomic) IBOutlet UIWebView *webView;

• (IBAction)btnDisplayFiles:(id)sender;

-(void)handleDocumentOpenURL:(NSURL )url; -(void)displayAlert:(NSString ) str;
-(void)loadFileFromDocumentsFolder:(NSString *) filename; -
(void)listFilesFromDocumentsFolder;

@end

// // ViewController.m // test // // Created by Alfred Jiang on 4/25/15. // Copyright
(c) 2015 Alfred Jiang. All rights reserved. //

```

# import "ViewController.h"

```
@interface ViewController ()
```

```
@end
```

```
@implementation ViewController
```

- `(void)viewDidLoad { [super viewDidLoad]; }`
- `(void)didReceiveMemoryWarning { [super didReceiveMemoryWarning]; // Dispose of any resources that can be recreated. }`
- `(IBAction)btnDisplayFiles:(id)sender { [self listFilesFromDocumentsFolder]; }`
- `(void)handleDocumentOpenURL:(NSURL )url { NSURLRequest requestObj = [NSURLRequest requestWithURL:url]; _webView setUserInteractionEnabled:YES]; _webView loadRequest:requestObj]; }`

```
-(void)loadFileFromDocumentsFolder:(NSString ) filename { ---get the path of the Documents folder---  
NSArray paths = NSSearchPathForDirectoriesInDomains( NSDocumentDirectory, NSUserDomainMask, YES); NSString documentsDirectory = [paths objectAtIndex:0]; NSString filePath = [documentsDirectory stringByAppendingPathComponent:filename]; NSURL *fileUrl = [NSURL fileURLWithPath:filePath]; [self handleDocumentOpenURL:fileUrl]; }
```

```
-(void)listFilesFromDocumentsFolder { ---get the path of the Documents folder---  
NSArray paths = NSSearchPathForDirectoriesInDomains( NSDocumentDirectory, NSUserDomainMask, YES); NSString documentsDirectory = [paths objectAtIndex:0];
```

```
NSFileManager *manager = [NSFileManager defaultManager];
NSArray *fileList =
[manager contentsOfDirectoryAtPath:documentsDirectory error:nil];
;
NSMutableString *filesStr =
NSMutableString stringWithString:@"Files in Documents folder \n";
];
for (NSString *s in fileList){
    [filesStr appendFormat:@"%@", s];
}

[self loadFileFromDocumentsFolder:@"ee.pdf"];

}
```

@end

#### ##### 3. 在其他应用中调用自己的应用打开系统支持的默认文档

1. 在 info.plist 中增加如下 字段

```
```xml
<key>CFBundleDocumentTypes</key>
<array>
<dict>
    <key>CFBundleTypeName</key>
    <string>PDF Document</string>
    <key>LSHandlerRank</key>
    <string>Alternate</string>
    <key>CFBundleTypeRole</key>
    <string>Viewer</string>
    <key>LSItemContentTypes</key>
    <array>
        <string>com.adobe.pdf</string>
    </array>
</dict>
</array>
```

|                         |            |              |
|-------------------------|------------|--------------|
| ▼ CFBundlDocumentTypes  | Array      | (1 item)     |
| ▼ Item 0 (PDF Document) | Dictionary | (4 items)    |
| CFBundlTypeName         | String     | PDF Document |
| LSHandlerRank           | String     | Alternate    |
| CFBundlTypeRole         | String     | Viewer       |
| ► LSItemContentTypes    | Array      | (1 item)     |

1. ViewController 代码实现如下 ``objectivec // // ViewController.h // test // //  
Created by Alfred Jiang on 4/25/15. // Copyright (c) 2015 Alfred Jiang. All  
rights reserved. //

## import

```
@interface ViewController : UIViewController

@property (weak, nonatomic) IBOutlet UIWebView *webView;

-(void)handleDocumentOpenURL:(NSURL *)url;

@end

// // ViewController.m // test // // Created by Alfred Jiang on 4/25/15. // Copyright
(c) 2015 Alfred Jiang. All rights reserved. //
```

## import "ViewController.h"

```
@interface ViewController ()

@end

@implementation ViewController @synthesize webView;

• -(void)viewDidLoad { [super viewDidLoad]; }

• -(void)didReceiveMemoryWarning { [super didReceiveMemoryWarning]; //
Dispose of any resources that can be recreated. }

• -(void)handleDocumentOpenURL:(NSURL )url { NSURLRequest requestObj =
[NSURLRequest requestWithURL:url]; [self.webView
setUserInteractionEnabled:YES]; [self.webView loadRequest:requestObj]; }

@end
```

3. 在 Appdelegate.m 中增加如下代码

```
```objective-c
-(BOOL)application:(UIApplication *)application openURL:(NSURL *)
"url sourceApplication:(NSString *)sourceApplication annotation:
(id)annotation {
    if (url != nil && [url isFileURL]) {
        [(ViewController *)self.window.rootViewController handle
DocumentOpenURL:url];
    }
    return YES;
}
```

1. 在支持“用其他应用打开”选项的应用中就可以看到自己的应用了



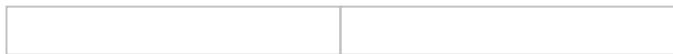
4. 在其他应用中调用自己的应用打开自定义文档

1. 在 info.plist 中增加如下字段

```
<key>CFBundleDocumentTypes</key>
<array>
<dict>
<key>CFBundleTypeName</key>
<string>Sudoku Game Document</string>
<key>LSHandlerRank</key>
<string>Owner</string>
<key>CFBundleTypeRole</key>
<string>Editor</string>
<key>LSItemContentTypes</key>
<array>
<string>net.learn2develop.offlinereader
.sdk</string>
</array>
</dict>
</array>
<key>UTEExportedTypeDeclarations</key>
<array>
<dict>
<key>UTTypeConformsTo</key>
<array>
<string>public.data</string>
</array>
<key>UTTypeTagSpecification</key>
<dict>
<key>public.filename-extension</key>
<string>testextension</string>
<key>public.mime-type</key>
<string>application/test</string>
</dict>
<key>UTTypeIdentifier</key>
<string>net.learn2develop.offlinereader.sdk
</string>
<key>UTTypeDescription</key>
<string>Sudoku Game Document</string>
</dict>
</array>
```

|                                           |            |                                     |
|-------------------------------------------|------------|-------------------------------------|
| ▼ Document types                          | Array      | (1 item)                            |
| ▼ Item 0 (Sudoku Game Document)           | Dictionary | (4 items)                           |
| Document Type Name                        | String     | Sudoku Game Document                |
| Handler rank                              | String     | Owner                               |
| Role                                      | String     | Editor                              |
| ▼ Document Content Type UTIs              | Array      | (1 item)                            |
| Item 0                                    | String     | net.learn2develop.offlinereader.sdk |
| ▼ Exported Type UTIs                      | Array      | (1 item)                            |
| ▼ Item 0 (net.learn2develop.offlinere...) | Dictionary | (4 items)                           |
| ▼ Conforms to UTIs                        | Array      | (1 item)                            |
| Item 0                                    | String     | public.data                         |
| ▼ Equivalent Types                        | Dictionary | (2 items)                           |
| public.filename-extension                 | String     | testextension                       |
| public.mime-type                          | String     | application/test                    |
| Identifier                                | String     | net.learn2develop.offlinereader.sdk |
| Description                               | String     | Sudoku Game Document                |

1. 在其他应用中选择“用其他应用打开”选项的应用中就可以看到自己的应用了



## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-05-07 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

应用间通信 - 通过 URL 检测是否安装并打开应用

## 关键字

应用间通信 \ URL \ Web \ 邮件打开App \ Schema

## 需求场景

1. 需要通过 URL 检测是否安装应用，如果已安装则打开应用，如果未安装则跳转到下载页面

## 参考链接

1. iOS使用schema协议调起APP
2. 在mobile safari中巧妙实现检测应用安装就打开，否则进App Store下载
3. CSDN - IOS在一个程序中启动另一个程序

## 详细内容

### 1. App 设置

在 .plist 文件添加如下字段

|                |       |            |               |
|----------------|-------|------------|---------------|
| ▼ URL types    | ◆ + - | Array      | (1 item)      |
| ▼ Item 0       |       | Dictionary | (2 items)     |
| URL identifier | ◆ + - | String     |               |
| ▼ URL Schemes  | ◆ + - | Array      | (1 item)      |
| Item 0         |       | String     | GJApplication |

```

<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleURLName</key>
    <string></string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>GJApplication</string>
    </array>
  </dict>
</array>

```

- GJApplication 替换为自定义名称

## 2. Web 代码

### 1. 示例一 ``html

Click to open GJ App  
[Open GJ App](#)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-04-14 | Alfred Jiang | -  |

## 方案名称

手势 - 判断点击位置是否在某区域 ( View ) 内

## 关键字

手势 \ 点击位置 \ 区域 \ UITapGestureRecognizer

## 需求场景

1. 需要对手势操作的点击位置进行判断时
2. 判断点击位置是否在某控件中时

## 参考链接

(无)

## 详细内容

判断 UITapGestureRecognizer 点击位置是否在 \_viewTest 控件中

方法一 : *CGRectContainsPoint*

```
- (void)touchMainViewGesture:(UITapGestureRecognizer *)gesture
{
    if (CGRectContainsPoint(_viewTest.bounds, [gesture locationInView:_viewTest])) {
        NSLog(@"%@", _viewTest touched);
    }
}
```

方法二 : - *(BOOL)pointInside:(CGPoint)point withEvent:(nullable UIEvent \*)event\**

```
- (void)touchMainViewGesture:(UITapGestureRecognizer *)gesture
{
    if ([_viewTest pointInside:[gesture locationInView:_viewTest]
] withEvent:nil]) {
        NSLog(@"%@", _viewTest touched);
    }
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-25 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

手势 - 实现手势操作介绍

## 关键字

手势 \ Gesture Recognizer \ 手势操作

## 需求场景

1. 需要对页面增加手势操作响应时

## 参考链接

1. [Apple documentation](#)
2. [iOS手势识别的详细使用：拖动、缩放、旋转、点击、手势依赖、自定义手势](#)
3. [UIGestureRecognizer Tutorial in iOS 5: Pinches, Pans, and More!](#)

## 详细内容

**SDK** 提供的手势

| 序号 | 手势                                 | Class                            | 说明                  |
|----|------------------------------------|----------------------------------|---------------------|
| 1  | Tap Gesture Recognizer             | UITapGestureRecognizer           | 点击手势                |
| 2  | Pinch Gesture Recognizer           | UIPinchGestureRecognizer         | 二指往内或往外拨动，平时经常用到的缩放 |
| 3  | Rotation Gesture Recognizer        | UIRotationGestureRecognizer      | 旋转手势                |
| 4  | Swipe Gesture Recognizer           | UISwipeGestureRecognizer         | 滑动，快速移动             |
| 5  | Pan Gesture Recognizer             | UIPanGestureRecognizer           | 拖移，慢速移动             |
| 6  | Screen Edge Pan Gesture Recognizer | UIScreenEdgePanGestureRecognizer | 屏幕边缘拖动手势            |
| 7  | Long Press Gesture Recognizer      | UILongPressGestureRecognizer     | 长按手势                |

## 自定义手势举例

### 定义 TickleGestureRecognizer

1. TickleGestureRecognizer.h ``objective-c

## import

```
typedef enum { DirectionUnknown = 0, DirectionLeft, DirectionRight } Direction;

@interface TickleGestureRecognizer : UIGestureRecognizer

@property (assign) int tickleCount; @property (assign) CGPoint curTickleStart;
@property (assign) Direction lastDirection;
```

@end

```
2. TickleGestureRecognizer.m
````objective-c
#import "TickleGestureRecognizer.h"
#import <UIKit/UIGestureRecognizerSubclass.h>

#define REQUIRED_TICKLES      2
#define MOVE_AMT_PER_TICKLE   25

@implementation TickleGestureRecognizer
@synthesize tickleCount;
@synthesize curTickleStart;
@synthesize lastDirection;

- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    UITouch * touch = [touches anyObject];
    self.curTickleStart = [touch locationInView:self.view];
}

- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
{

    // Make sure we've moved a minimum amount since curTickleStart
    UITouch * touch = [touches anyObject];
    CGPoint ticklePoint = [touch locationInView:self.view];
    CGFloat moveAmt = ticklePoint.x - curTickleStart.x;
    Direction curDirection;
    if (moveAmt < 0) {
        curDirection = DirectionLeft;
    } else {
        curDirection = DirectionRight;
    }
    if (ABS(moveAmt) < MOVE_AMT_PER_TICKLE) return;

    // Make sure we've switched directions
    if (self.lastDirection == DirectionUnknown ||
```

```
        (self.lastDirection == DirectionLeft && curDirection ==  
DirectionRight) ||  
        (self.lastDirection == DirectionRight && curDirection ==  
DirectionLeft)) {  
  
    // w00t we've got a tickle!  
    self.tickleCount++;  
    self.curTickleStart = ticklePoint;  
    self.lastDirection = curDirection;  
  
    // Once we have the required number of tickles, switch t  
he state to ended.  
    // As a result of doing this, the callback will be calle  
d.  
    if (self.state == UIGestureRecognizerStatePossible && se  
lf.tickleCount > REQUIRED_TICKLES) {  
        [self setState:UIGestureRecognizerStateEnded];  
    }  
}  
  
}  
  
- (void)resetState {  
    self.tickleCount = 0;  
    self.curTickleStart = CGPointMakeZero;  
    self.lastDirection = DirectionUnknown;  
    if (self.state == UIGestureRecognizerStatePossible) {  
        [self setState:UIGestureRecognizerStateFailed];  
    }  
}  
  
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event  
{  
    [self resetState];  
}  
  
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)e  
vent  
{  
    [self resetState];
```

```
}
```

```
@end
```

## 使用 TickleGestureRecognizer

1. ViewController.h ```objectivec // Add to top of file

# import "TickleGestureRecognizer.h"

```
// Add after @interface @property (strong) AVAudioPlayer * hehePlayer;
```

- (void)handleTickle:(TickleGestureRecognizer \*)recognizer; ```
- ViewController.m ```objectivec // After @implementation @synthesize hehePlayer;

```
// In viewDidLoad, right after TODO TickleGestureRecognizer * recognizer2 = [[TickleGestureRecognizer alloc] initWithTarget:self action:@selector(handleTickle:)]; recognizer2.delegate = self; [view addGestureRecognizer:recognizer2];
```

```
// At end of viewDidLoad self.hehePlayer = [self loadWav:@"hehehe1"];
```

```
// Add at beginning of handlePan (gotta turn off pan to recognize tickles) return;
```

```
// At end of file
```

- (void)handleTickle:(TickleGestureRecognizer \*)recognizer { [self.hehePlayer play]; } ```

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

授权 - 使用 ClusterPrePermissions 更加友好的提示授权操作

## 关键字

授权 \ 通讯录 \ 相册 \ 地址

## 需求场景

1. 需要用户授权通讯录、相册、地址等信息时提前增加提示

## 参考链接

1. [GitHub - ClusterPrePermissions](#)

## 详细内容

使用方法

```
ClusterPrePermissions *permissions = [ClusterPrePermissions sharedPermissions];
[permissions showPhotoPermissionsWithTitle:@"Access your photos?"

                                         message:@"Your message here"
                                         denyButtonTitle:@"Not Now"
                                         grantButtonTitle:@"Give Access"
                                         completionHandler:^(BOOL hasPermission,
                                                               ClusterDialogResult
                                                               userDialogResult,
                                                               ClusterDialogResult
                                                               systemDialogResult) {
                                         if (hasPermission) {
                                             // Continue with your code here
                                         } else {
                                             // Handle access not being available
                                         }
                                         }];

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-02-27 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

数据加密 - 3DES 加密(包含JAVA服务器端解密)

## 关键字

数据加密 \ 加密 \ 解密 \ 3DES

## 需求场景

1. 移动端与服务器敏感数据通讯
2. 移动端本地部分需要保存的敏感数据（NSUserDefaults、文件或数据库）

## 参考链接

1. [CSDN - iOS 3DES加密 和 java 3DES 解密](#)

## 详细内容

### 1. 首先进入头文件,添加必要的 **Framework**

```
#import <CommonCrypto/CommonDigest.h>
#import <CommonCrypto/CommonCryptor.h>
#import <Security/Security.h>
```

### 2. iOS 端实现代码

```
#define kChosenDigestLength CC_SHA1_DIGEST_LENGTH
```

```

#define DESKEY @"D6D2402F1C98E208FF2E863AA29334BD65AE1932A821502
D9E5673CDE3C713ACFE53E2103CD40ED6BEBB101B484CAE83D537806C6CB611A
EE86ED2CA8C97BBE95CF8476066D419E8E833376B850172107844D394016715B
2E47E0A6EECB3E83A361FA75FA44693F90D38C6F62029FCD8EA395ED868F9D71
8293E9C0E63194E87"

-(NSString*)TripleDES:(NSString*)plainText encryptOrDecrypt:(CCO
peration)encryptOrDecrypt
{

const void *vplainText;
size_t plainTextBufferSize;

if (encryptOrDecrypt == kCCDecrypt)//解密
{
    NSData *EncryptData = [GTMBase64 decodeData:[plainText dataU
singEncoding:NSUTF8StringEncoding]];
    plainTextBufferSize = [EncryptData length];
    vplainText = [EncryptData bytes];
}
else //加密
{
    NSData* data = [plainText dataUsingEncoding:NSUTF8StringEncoding];
    plainTextBufferSize = [data length];
    vplainText = (const void *)[data bytes];
}

CCCryptorStatus ccStatus;
uint8_t *bufferPtr = NULL;
size_t bufferPtrSize = 0;
size_t movedBytes = 0;

bufferPtrSize = (plainTextBufferSize + kCCBlockSize3DES) & ~(kCC
BlockSize3DES - 1);
bufferPtr = malloc( bufferPtrSize * sizeof(uint8_t));
memset((void *)bufferPtr, 0x0, bufferPtrSize);
// memset((void *) iv, 0x0, (size_t) sizeof(iv));

```

```

const void *vkey = (const void *)[DESKEY UTF8String];
// NSString *initVec = @"init Vec";
//const void *vinitVec = (const void *) [initVec UTF8String];
// Byte iv[] = {0x12, 0x34, 0x56, 0x78, 0x90, 0xAB, 0xCD, 0xEF};

ccStatus = CCCrypt(encryptOrDecrypt,
                    kCCHAlgorithm3DES,
                    kCCOptionPKCS7Padding | kCCOptionECBMode,
                    vkey,
                    kCCKeySize3DES,
                    nil,
                    vplainText,
                    plainTextBufferSize,
                    (void *)bufferPtr,
                    bufferPtrSize,
                    &movedBytes);

//if (ccStatus == kCCSuccess) NSLog(@"SUCCESS");
/*else if (ccStatus == kCC ParamError) return @"PARAM ERROR";
 else if (ccStatus == kCCBufferTooSmall) return @"BUFFER TOO SMA
LL";
 else if (ccStatus == kCCMemoryFailure) return @"MEMORY FAILURE"
;
 else if (ccStatus == kCCAlignmentError) return @"ALIGNMENT";
 else if (ccStatus == kCCDecodeError) return @"DECODE ERROR";
 else if (ccStatus == kCCUnimplemented) return @"UNIMPLEMENTED";
 */

```

**NSString** \*result;

```

if (encryptOrDecrypt == kCCDecrypt)
{
    result = [[[NSString alloc] initWithData:[NSData dataWithByt
es:(const void *)bufferPtr
                           leng
th:(NSUInteger)movedBytes]
                                         encoding:NSUTF8StringEncoding
]
                           autorelease];
}
else

```

```

{
    NSData *myData = [NSData dataWithBytes:(const void *)bufferP
tr length:(NSUInteger)movedBytes];
    result = [GTMBase64 stringByEncodingData:myData];
}

return result;
}

```

### 3. Java 端 3DES 解密和加密算法

```

public static String encryptThreeDESECB(String src, String key) t
hrows Exception
{
    DESedeKeySpec dks = new DESedeKeySpec(key.getBytes("UTF-8"))
;
    SecretKeyFactory keyFactory = SecretKeyFactory.getInstance("D
ESede");
    SecretKey securekey = keyFactory.generateSecret(dks);

    Cipher cipher = Cipher.getInstance("DESede/ECB/PKCS5Padding"
);
    cipher.init(Cipher.ENCRYPT_MODE, securekey);
    byte[] b=cipher.doFinal(src.getBytes());

    BASE64Encoder encoder = new BASE64Encoder();
    return encoder.encode(b).replaceAll("\r", "").replaceAll("\n"
, "");
}

//3DESECB解密, key必须是长度大于等于 3*8 = 24 位
public static String decryptThreeDESECB(String src, String key) t
hrows Exception
{
    //--通过base64, 将字符串转成byte数组
    BASE64Decoder decoder = new BASE64Decoder();
    byte[] bytesrc = decoder.decodeBuffer(src);
    //--解密的key
}

```

```
DESedeKeySpec dks = new DESedeKeySpec(key.getBytes("UTF-8"))
;
SecretKeyFactory keyFactory = SecretKeyFactory.getInstance("DESede");
SecretKey securekey = keyFactory.generateSecret(dks);

//--Cipher对象解密
Cipher cipher = Cipher.getInstance("DESede/ECB/PKCS5Padding");
cipher.init(Cipher.DECRYPT_MODE, securekey);
byte[] retByte = cipher.doFinal(bytesrc);

return new String(retByte);
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-02-27 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

数据加密 - AES 加密解密

## 关键字

数据加密 \ 加密 \ 解密 \ AES

## 需求场景

1. 移动端与服务器敏感数据通讯加密需求
2. 移动端本地部分需要保存的敏感数据（NSUserDefaults、文件或数据库）

## 参考链接

1. [GitHub - NSData+AES.h](#)
2. [GitHub - RNCryptor/RNCryptor](#):RNCryptor是一个跨语言的AES加密/解密数据格式，最初针对Objective-C，现在已可以使用C、Java、PHP、Python以及Ruby语言。该数据格式包含AES加密所需的所有元数据，同时支持同步和异步读取加密文件，功能十分强大，并且还有详细的文档。

## 详细内容

代码实现如下

NSData+AES.h文件

```
/***
http://mythosil.hatenablog.com/entry/20111017/1318873155
http://blog.dealforest.net/2012/03/ios-android-per-aes-crypt-
connection/
*/

@interface NSData (AES)

- (NSData *)AES128EncryptedDataWithKey:(NSString *)key;
- (NSData *)AES128DecryptedDataWithKey:(NSString *)key;
- (NSData *)AES128EncryptedDataWithKey:(NSString *)key iv:(NSStr-
ing *)iv;
- (NSData *)AES128DecryptedDataWithKey:(NSString *)key iv:(NSStr-
ing *)iv;

@end
```

### NSData+AES.m文件

```
#import "NSData+AES.h"
#import <CommonCrypto/CommonCryptor.h>

@implementation NSData (AES)

- (NSData *)AES128EncryptedDataWithKey:(NSString *)key
{
    return [self AES128EncryptedDataWithKey:key iv:nil];
}

- (NSData *)AES128DecryptedDataWithKey:(NSString *)key
{
    return [self AES128DecryptedDataWithKey:key iv:nil];
}

- (NSData *)AES128EncryptedDataWithKey:(NSString *)key iv:(NSStr-
ing *)iv
{
    return [self AES128Operation:kCCEncrypt key:key iv:iv];
}
```

```

- (NSData *)AES128DecryptedDataWithKey:(NSString *)key iv:(NSString *)iv
{
    return [self AES128Operation:kCCDecrypt key:key iv:iv];
}

- (NSData *)AES128Operation:(CCOperation)operation key:(NSString *)key iv:(NSString *)iv
{
    char keyPtr[kCCKeySizeAES128 + 1];
    bzero(keyPtr, sizeof(keyPtr));
    [key getCString:keyPtr maxLength:sizeof(keyPtr) encoding:NSUTF8StringEncoding];

    char ivPtr[kCCBlockSizeAES128 + 1];
    bzero(ivPtr, sizeof(ivPtr));
    if (iv) {
        [iv getCString:ivPtr maxLength:sizeof(ivPtr) encoding:NSUTF8StringEncoding];
    }

    NSUInteger dataLength = [self length];
    size_t bufferSize = dataLength + kCCBlockSizeAES128;
    void *buffer = malloc(bufferSize);

    size_t numBytesEncrypted = 0;
    CCCryptorStatus cryptStatus = CCCrypt(operation,
                                            kCCAlgorithmAES128,
                                            kCCOptionPKCS7Padding |
                                            kCCOptionECBMode,
                                            keyPtr,
                                            kCCBlockSizeAES128,
                                            ivPtr,
                                            [self bytes],
                                            dataLength,
                                            buffer,
                                            bufferSize,
                                            &numBytesEncrypted);
    if (cryptStatus == kCCSuccess) {

```

```
        return [NSData dataWithBytesNoCopy:buffer length:numBytesEncrypted];
    }
    free(buffer);
    return nil;
}

@end
```

## 效果图

(无)

## 备注

这里AES在iOS加过密以后以NSData的形式存下来，如果想以NSString形式存储，那么对NSData进行base64位编码。参考

- [数据加密 - Base64 编码 \( NSData <=> NSString \)](#)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-02-27 | Alfred Jiang | -  |
| 2  | 2015-12-18 | Alfred Jiang | -  |

## 方案名称

数据加密 - Base64 编码 ( NSData <=> NSString )

## 关键字

数据加密 \ 编码 \ 解码 \ Base64 \ NSData \ NSString \ NSData <=> NSString

## 需求场景

1. 需要将 `NSData` 类型数据转为 `NSString` 类型数据时
2. 对 `NSString` 或 `NSData` 有 Base64 编码需求时

## 参考链接

1. [GitHub - google-toolbox-for-mac](#)
2. [CSDN - Base64编码介绍](#)

## 详细内容

iOS 实现 **Base64** 编码主要通过 [Google Toolbox for Mac](#) 中的 **GTMBase64** 相关文件实现。

1. 下载 [Google Toolbox for Mac](#) 工程文件，将该工程中以下三个文件引入自己工程中 ``objectivec GTMDefines.h

`GTMBase64.h`

`GTMBase64.m`

2. 实现如下编码解码方法

.h文件

```
```objectiveC
#pragma mark - base64
+ (NSString*)encodeBase64String:(NSString *)input;
+ (NSString*)decodeBase64String:(NSString *)input;
+ (NSData*)encodeBase64Data:(NSData *)data;
+ (NSData*)decodeBase64Data:(NSData *)data;
```

.m文件

```

#pragma mark - base64
+ (NSString *)encodeBase64String:(NSString * )input {
    NSData *data = [input dataUsingEncoding:NSUTF8StringEncoding
allowLossyConversion:YES];
    data = [GTMBase64 encodeData:data];
    NSString *base64String = [[[NSString alloc] initWithData:data
encoding:NSUTF8StringEncoding] autorelease];
    return base64String;
}

+ (NSString *)decodeBase64String:(NSString * )input {
    NSData *data = [input dataUsingEncoding:NSUTF8StringEncoding
allowLossyConversion:YES];
    data = [GTMBase64 decodeData:data];
    NSString *base64String = [[[NSString alloc] initWithData:data
encoding:NSUTF8StringEncoding] autorelease];
    return base64String;
}

+ (NSString *)encodeBase64Data:(NSData *)data {
    data = [GTMBase64 encodeData:data];
    NSString *base64String = [[[NSString alloc] initWithData:data
encoding:NSUTF8StringEncoding] autorelease];
    return base64String;
}

+ (NSString *)decodeBase64Data:(NSData *)data {
    data = [GTMBase64 decodeData:data];
    NSString *base64String = [[[NSString alloc] initWithData:data
encoding:NSUTF8StringEncoding] autorelease];
    return base64String;
}

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-02-27 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

数据加密 - DES 加密(包含PHP服务器端解密)

## 关键字

数据加密 \ 加密 \ 解密 \ DES

## 需求场景

- 移动端与服务器敏感数据通讯
- 移动端本地部分需要保存的敏感数据（NSUserDefaults、文件或数据库）

## 参考链接

- iOS Objective-C 與 PHP DES 加解密演算法實作

## 详细内容

### 1. 首先进入头文件,添加必要的 **Framework**

```
#import <CommonCrypto/CommonCryptor.h>
```

### 2. iOS 端实现代码

```
//解密
-(NSString*) decryptUseDES:(NSString*)cipherText key:(NSString*)
key {
```

```

// 利用 GTMBase64 解码 Base64 字串
NSData* cipherData = [GTMBase64 decodeString:cipherText];
unsigned char buffer[1024];
memset(buffer, 0, sizeof(char));
size_t numBytesDecrypted = 0;

// IV 偏移量不需使用
CCCryptorStatus cryptStatus = CCCrypt(kCCDecrypt,
                                         kCCAlgorithmDES,
                                         kCCOptionPKCS7Padding
                                         | kCCOptionECBMode,
                                         [key UTF8String],
                                         kCCKeySizeDES,
                                         nil,
                                         [cipherData bytes],
                                         [cipherData length],
                                         buffer,
                                         1024,
                                         &numBytesDecrypted);

NSString* plainText = nil;
if (cryptStatus == kCCSuccess) {
    NSData* data = [NSData dataWithBytes:buffer length:(NSInteger)numBytesDecrypted];
    plainText = [[[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding] autorelease];
}
return plainText;
}

//加密
-(NSString *) encryptUseDES:(NSString *)clearText key:(NSString *)key
{
    NSData *data = [clearText dataUsingEncoding:NSUTF8StringEncoding allowLossyConversion:YES];
    unsigned char buffer[1024];
    memset(buffer, 0, sizeof(char));
    size_t numBytesEncrypted = 0;

    CCCryptorStatus cryptStatus = CCCrypt(kCCEncrypt,

```

```

        KCAlgorithmDES,
        kCCOptionPKCS7Padding
| kCCOptionECBMode,
        [key UTF8String],
        kCCKeySizeDES,
        nil,
        [data bytes],
        [data length],
        buffer,
        1024,
&numBytesEncrypted);

NSString* plainText = nil;
if (cryptStatus == kCCSuccess) {
    NSData *dataTemp = [NSData dataWithBytes:buffer length:(NSUInteger)numBytesEncrypted];
    plainText = [GTMBase64 stringByEncodingData:dataTemp];
}else{
    NSLog(@"%@", @"DES加密失败");
}
return plainText;
}

```

### 3. PHP 端 DES 解密和加密算法

```

<?php
/**
 * PHP DES 加密程式
 *
 * @param $key 密鑰 (八個字元內)
 * @param $encrypt 要加密的明文
 * @return string 密文
 */
function encrypt ($key, $encrypt)
{
    // 根據 PKCS#7 RFC 5652 Cryptographic Message Syntax (CMS) 修
    正 Message 加入 Padding
    $block = mcrypt_get_block_size(MCRYPT_DES, MCRYPT_MODE_ECB);
    $pad = $block - (strlen($encrypt) % $block);
}

```

```

$encrypt .= str_repeat(chr($pad), $pad);

// 不需要設定 IV 進行加密
$passcrypt = mcrypt_encrypt(MCRYPT_DES, $key, $encrypt, MCRYPT_MODE_ECB);
return base64_encode($passcrypt);
}

/**
 * PHP DES 解密程式
 *
 * @param $key 密鑰 (八個字元內)
 * @param $decrypt 要解密的密文
 * @return string 明文
 */
function decrypt ($key, $decrypt)
{
    // 不需要設定 IV
    $str = mcrypt_decrypt(MCRYPT_DES, $key, base64_decode($decrypt), MCRYPT_MODE_ECB);

    // 根據 PKCS#7 RFC 5652 Cryptographic Message Syntax (CMS) 修正 Message 移除 Padding
    $pad = ord($str[strlen($str) - 1]);
    return substr($str, 0, strlen($str) - $pad);
}

$key = 'skey';
$plain = '0123ABCD!@#$中文';
$encrypt = encrypt($key, $plain);
$decrypt = decrypt($key, $encrypt);

echo 'plain = ' . $plain . "\n";
echo 'encrypt = ' . $encrypt . "\n";
echo 'decrypt = ' . $decrypt . "\n";
?>

```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-02-27 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

数据加密 - MD5 加密

## 关键字

数据加密 \ 加密 \ 解密 \ MD5 \ 摘要算法

## 需求场景

1. 对信息有特殊的 MD5 摘要算法需求

## 参考链接

1. [Stack Overflow](#)

## 详细内容

NSString+MyAdditions.h

```
@interface NSString (MyAdditions)
- (NSString *)md5;
@end
```

NSData+MyAdditions.h

```
@interface NSData (MyAdditions)
- (NSString *)md5;
@end
```

### NSString+MyAdditions.m

```
#import "MyAdditions.h"
#import <CommonCrypto/CommonDigest.h> // Need to import for CC_MD5 access

@implementation NSString (MyAdditions)
- (NSString *)md5
{
    const char *cStr = [self UTF8String];
    unsigned char result[CC_MD5_DIGEST_LENGTH];
    CC_MD5( cStr, strlen(cStr), result ); // This is the md5 call

    return [NSString stringWithFormat:
        @"%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x",
        result[0], result[1], result[2], result[3],
        result[4], result[5], result[6], result[7],
        result[8], result[9], result[10], result[11],
        result[12], result[13], result[14], result[15]
    ];
}
@end
```

### NSData+MyAdditions.m

```
@implementation NSData (MyAdditions)
- (NSString*)md5
{
    unsigned char result[CC_MD5_DIGEST_LENGTH];
    CC_MD5( self.bytes, self.length, result ); // This is the md5 call
    return [NSString stringWithFormat:
        @"%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x",
        result[0], result[1], result[2], result[3],
        result[4], result[5], result[6], result[7],
        result[8], result[9], result[10], result[11],
        result[12], result[13], result[14], result[15]
    ];
}
@end
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-12 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

数据加密 - RSA 加密(包含JAVA服务器端解密)

## 关键字

数据加密 \ 加密 \ 解密 \ RSA \ 非对称加密算法

## 需求场景

1. 移动端与服务器敏感数据通讯加密需求

## 参考链接

1. [ITEYE - iOS中使用RSA对数据进行加密解密](#)
2. [GitHub - gen\\_rsakey.sh](#)
3. [GitHub - RSAEncryptor](#)

## 详细内容

1. 使用**openssl**生成密匙对

`gen_rsakey.sh`

```
#!/usr/bin/env bash
echo "Generating RSA key pair ..."
echo "1024 RSA key: private_key.pem"
openssl genrsa -out private_key.pem 1024

echo "create certification require file: rsaCertReq.csr"
openssl req -new -key private_key.pem -out rsaCertReq.csr

echo "create certification using x509: rsaCert.crt"
openssl x509 -req -days 3650 -in rsaCertReq.csr -signkey private
_key.pem -out rsaCert.crt

echo "create public_key.der For IOS"
openssl x509 -outform der -in rsaCert.crt -out public_key.der

echo "create private_key.p12 For IOS. Please remember your passw
ord. The password will be used in iOS."
openssl pkcs12 -export -out private_key.p12 -inkey private_key.p
em -in rsaCert.crt

echo "create rsa_public_key.pem For Java"
openssl rsa -in private_key.pem -out rsa_public_key.pem -pubout
echo "create pkcs8_private_key.pem For Java"
openssl pkcs8 -topk8 -in private_key.pem -out pkcs8_private_key.
 pem -nocrypt

echo "finished."
```

## 2. 导入工具类 RSAEncryptor

RSAEncryptor.h

```
//  
//  RSAEncryptor.h  
//  RSATestProjct  
//  
//  Created by Alfred Jiang on 3/12/15.  
//  Copyright (c) 2015 Alfred Jiang. All rights reserved.  
  
#import <Foundation/Foundation.h>  
  
@interface RSAEncryptor : NSObject  
  
#pragma mark - Instance Methods  
  
-(void) loadPublicKeyFromFile: (NSString*) derFilePath;  
-(void) loadPublicKeyFromData: (NSData*) derData;  
  
-(void) loadPrivateKeyFromFile: (NSString*) p12FilePath password  
:(NSString*)p12Password;  
-(void) loadPrivateKeyFromData: (NSData*) p12Data password:(NSString*)p12Password;  
  
-(NSString*) rsaEncryptString:(NSString*)string;  
-(NSData*) rsaEncryptData:(NSData*)data ;  
  
-(NSString*) rsaDecryptString:(NSString*)string;  
-(NSData*) rsaDecryptData:(NSData*)data;  
  
-(BOOL) rsaSHA1VerifyData:(NSData *) plainData  
withSignature:(NSData *) signature;  
  
#pragma mark - Class Methods  
  
+(void) setSharedInstance: (RSAEncryptor*)instance;  
+(RSAEncryptor*) sharedInstance;  
  
@end
```

```
//  
//  RSAEncryptor.m  
//  RSAProjct  
//  
//  Created by Alfred Jiang on 3/12/15.  
//  Copyright (c) 2015 Alfred Jiang. All rights reserved.  
  
  
#import <Security/Security.h>  
#import "RSAEncryptor.h"  
#import "GTMBase64.h"  
#import <CommonCrypto/CommonCrypto.h>  
  
@implementation RSAEncryptor  
{  
    SecKeyRef publicKey;  
    SecKeyRef privateKey;  
}  
  
-(void)dealloc  
{  
    if (nil != publicKey) {  
        CFRelease(publicKey);  
    }  
    if (nil != privateKey) {  
        CFRelease(privateKey);  
    }  
}  
  
-(SecKeyRef) getPublicKey {  
    return publicKey;  
}  
  
-(SecKeyRef) getPrivateKey {  
    return privateKey;  
}  
  
-(void) loadPublicKeyFromFile: (NSString*) derFilePath  
{
```

```

        NSData *derData = [[NSData alloc] initWithContentsOfFile:der
FilePath];
        [self loadPublicKeyFromData: derData];
}
-(void) loadPublicKeyFromData: (NSData*) derData
{
    publicKey = [self getPublicKeyRefrenceFromeData: derData];
}

-(void) loadPrivateKeyFromFile: (NSString*) p12FilePath password
:(NSString*)p12Password
{
    NSData *p12Data = [NSData dataWithContentsOfFile:p12FilePath];
    [self loadPrivateKeyFromData: p12Data password:p12Password];
}

-(void) loadPrivateKeyFromData: (NSData*) p12Data password:(NSString*)
p12Password
{
    privateKey = [self getPrivateKeyRefrenceFromData: p12Data pa
ssword: p12Password];
}

#pragma mark - Private Methods

-(SecKeyRef) getPublicKeyRefrenceFromeData: (NSData*)derData
{
    SecCertificateRef myCertificate = SecCertificateCreateWithData(
kCFAllocatorDefault, (__bridge CFDataRef)derData);
    SecPolicyRef myPolicy = SecPolicyCreateBasicX509();
    SecTrustRef myTrust;
    OSStatus status = SecTrustCreateWithCertificates(myCertifica
te,myPolicy,&myTrust);
    SecTrustResultType trustResult;
    if (status == noErr) {
        status = SecTrustEvaluate(myTrust, &trustResult);
    }
    SecKeyRef securityKey = SecTrustCopyPublicKey(myTrust);
    CFRelease(myCertificate);
}

```

```

    CFRelease(myPolicy);
    CFRelease(myTrust);

    return securityKey;
}

-(SecKeyRef) getPrivateKeyRefrenceFromData: (NSData*)p12Data pas
sword:(NSString*)password
{
    SecKeyRef privateKeyRef = NULL;
    NSMutableDictionary * options = [[NSMutableDictionary alloc]
init];
    [options setObject: password forKey:(__bridge id)kSecImportE
xportPassphrase];
    CFArrayRef items = CFArrayCreate(NULL, 0, 0, NULL);
    OSStatus securityError = SecPKCS12Import((__bridge CFDataRef
) p12Data, (__bridge CFDictionaryRef)options, &items);
    if (securityError == noErr && CFArrayGetCount(items) > 0) {
        CFDictionaryRef identityDict = CFArrayGetValueAtIndex(it
ems, 0);
        SecIdentityRef identityApp = (SecIdentityRef)CFDictionaryar
yGetValue(identityDict, kSecImportItemIdentity);
        securityError = SecIdentityCopyPrivateKey(identityApp, &
privateKeyRef);
        if (securityError != noErr) {
            privateKeyRef = NULL;
        }
    }
    CFRelease(items);

    return privateKeyRef;
}

#pragma mark - Encrypt

-(NSString*) rsaEncryptString:(NSString*)string {
    NSData* data = [string dataUsingEncoding:NSUTF8StringEncoding
];
    NSData* encryptedData = [self rsaEncryptData: data];
    NSString* base64EncryptedString = [GTMBase64 stringByEncodin

```

```

gData:encryptedData];
//      [encryptedData base64EncodedString];
    return base64EncryptedString;
}

// 加密的大小受限于SecKeyEncrypt函数，SecKeyEncrypt要求明文和密钥的长度一致，如果要加密更长的内容，需要把内容按密钥长度分成多份，然后多次调用SecKeyEncrypt来实现
-(NSData*) rsaEncryptData:(NSData*)data {
    SecKeyRef key = [self getPublicKey];
    size_t cipherBufferSize = SecKeyGetBlockSize(key);
    uint8_t *cipherBuffer = malloc(cipherBufferSize * sizeof(uint8_t));
    size_t blockSize = cipherBufferSize - 11;           // 分段加密
    size_t blockCount = (size_t)ceil([data length] / (double)blockSize);
    NSMutableData *encryptedData = [[NSMutableData alloc] init];
    for (int i=0; i<blockCount; i++) {
        int bufferSize = MIN(blockSize,[data length] - i * blockSize);
        NSData *buffer = [data subdataWithRange:NSMakeRange(i * blockSize, bufferSize)];
        OSStatus status = SecKeyEncrypt(key, kSecPaddingPKCS1, (const uint8_t *)[buffer bytes], [buffer length], cipherBuffer, &cipherBufferSize);
        if (status == noErr){
            NSData *encryptedBytes = [[NSData alloc] initWithBytes:(const void *)cipherBuffer length:cipherBufferSize];
            [encryptedData appendData:encryptedBytes];
        }else{
            if (cipherBuffer) {
                free(cipherBuffer);
            }
            return nil;
        }
    }
    if (cipherBuffer){
        free(cipherBuffer);
    }
}

```

```
    return encryptedData;
}

#pragma mark - Decrypt

-(NSString*) rsaDecryptString:(NSString*)string {

    NSData* data = [[NSData alloc] initWithBase64EncodedString:string options:NSDataBase64DecodingIgnoreUnknownCharacters];
    NSData* decryptData = [self rsaDecryptData: data];
    NSString* result = [[NSString alloc] initWithData: decryptData encoding:NSUTF8StringEncoding];
    return result;
}

-(NSData*) rsaDecryptData:(NSData*)data {
    SecKeyRef key = [self getPrivateKey];
    size_t cipherLen = [data length];
    void *cipher = malloc(cipherLen);
    [data getBytes:cipher length:cipherLen];
    size_t plainLen = SecKeyGetBlockSize(key) - 12;
    void *plain = malloc(plainLen);
    OSStatus status = SecKeyDecrypt(key, kSecPaddingPKCS1, cipher, cipherLen, plain, &plainLen);

    if (status != noErr) {
        return nil;
    }

    NSData *decryptedData = [[NSData alloc] initWithBytes:(const void *)plain length:plainLen];

    return decryptedData;
}

#pragma mark - verify file SHA1
-(BOOL) rsaSHA1VerifyData:(NSData *) plainData
                      withSignature:(NSData *) signature {

    size_t signedHashBytesSize = SecKeyGetBlockSize([self getPub
```

```

licKey]);
    const void* signedHashBytes = [signature bytes];

    size_t hashBytesSize = CC_SHA1_DIGEST_LENGTH;
    uint8_t* hashBytes = malloc(hashBytesSize);
    if (!CC_SHA1([plainData bytes], (CC_LONG)[plainData length],
    hashBytes)) {
        return NO;
    }

    OSStatus status = SecKeyRawVerify(publicKey,
                                      kSecPaddingPKCS1SHA1,
                                      hashBytes,
                                      hashBytesSize,
                                      signedHashBytes,
                                      signedHashBytesSize);

    return status == errSecSuccess;
}

#pragma mark - Class Methods

static RSAEncryptor* sharedInstance = nil;

+(void) setSharedInstance: (RSAEncryptor*)instance
{
    sharedInstance = instance;
}

+(RSAEncryptor*) sharedInstance
{
    return sharedInstance;
}

@end

```

### 3. 使用 **RSAEncryptor** 进行**RSA**加密解密

```

RSAEncryptor *rsa = [[RSAEncryptor alloc] init];

NSLog(@"encryptor using rsa");
NSString *publicKeyPath = [[NSBundle mainBundle] pathForResource:@"public_key" ofType:@"der"];
NSLog(@"public key: %@", publicKeyPath);
[rsa loadPublicKeyFromFile:publicKeyPath];

NSString *securityText = @"hello ~";
NSString *encryptedString = [rsa rsaEncryptString:securityText];
NSLog(@"encrypted data: %@", encryptedString);

NSLog(@"decryptor using rsa");
[rsa loadPrivateKeyFromFile:[[NSBundle mainBundle] pathForResource:@"private_key" ofType:@"p12"] password:@"1234"];
NSString *decryptedString = [rsa rsaDecryptString:encryptedString];
NSLog(@"decrypted data: %@", decryptedString);

```

#### 4. 在服务器端解码数据(Java)

##### 1. 生成的pkcs8 private key

```
openssl pkcs8 -topk8 -in private_key.pem -out pkcs8_private_key.pem -nocrypt
```

##### 2. 具体解码步骤 `` 加载pkcs8 private key: 读取private key文件 去掉private key头尾的"-----BEGIN PRIVATE KEY-----"和"-----END PRIVATE KEY-----" 删除 private key中的换行 对处理后的数据进行Base64解码 使用解码后的数据生成 private key.

解密数据: 对数据进行Base64解码 使用RSA decrypt数据.

##### 3. 实现代码

```
```objective-c
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
```

```
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import java.io.IOException;
import java.nio.charset.Charset;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.security.InvalidKeyException;
import java.security.KeyFactory;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.PKCS8EncodedKeySpec;
import java.util.Base64;

import static java.lang.String.format;

public class Encryptor {

    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeySpecException, NoSuchPaddingException, InvalidKeyException, BadPaddingException, IllegalBlockSizeException {
        PrivateKey privateKey = readPrivateKey();

        String message = "AFppaFPTbmboMZD55cjCfrVaWUW7+hZkaq160d+6fP0lwz/yC+Rshb/8cf5BpBlUao2EunchnzeKxzpiPqtCcCITKvk6HcFKZS0sN9w0hlQFYT+I4f/CZITwBVAJaldZ7mky0iuvM+raXMwrS+7MLKgYXkd5cFPxEsTxpMSa5Nk=";
        System.out.println(format("- decrypt rsa encrypted base64 message: %s", message));
        // hello ~, encrypted and encoded with Base64:
        byte[] data = encryptedData(message);
        String text = decrypt(privateKey, data);
        System.out.println(text);
    }

    private static String decrypt(PrivateKey privateKey, byte[] data) throws NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException, IllegalBlockSizeException, BadPaddingException {

```

```
Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
cipher.init(Cipher.DECRYPT_MODE, privateKey);
byte[] decryptedData = cipher.doFinal(data);

return new String(decryptedData);
}

private static byte[] encryptedData(String base64Text) {
    return Base64.getDecoder().decode(base64Text.getBytes(
        Charset.forName("UTF-8")));
}

private static PrivateKey readPrivateKey() throws IOException,
    NoSuchAlgorithmException, InvalidKeySpecException {
    byte[] privateKeyData = Files.readAllBytes(
        Paths.get("/Users/twer/macspace/ios_workshop/Security/SecurityLogin/tools/pkcs8_private_key.pem"));

    byte[] decodedKeyData = Base64.getDecoder()
        .decode(new String(privateKeyData)
            .replaceAll("----\w+ PRIVATE KEY----"
            , ""))
        .replace("\n", "")
        .getBytes();

    return KeyFactory.getInstance("RSA").generatePrivate(new
        PKCS8EncodedKeySpec(decodedKeyData));
}
}
```

## 效果图

(无)

## 备注

RSA使用"秘匙对"对数据进行加密解密.在加密解密数据前,需要先生成公钥(**public key**)和私钥(**private key**).

公钥(**public key**): 用于加密数据. 用于公开, 一般存放在数据提供方, 例如iOS客户端.

私钥(**private key**): 用于解密数据. 必须保密, 私钥泄露会造成安全问题.

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-18 | Alfred Jiang | -  |

## 方案名称

数据加密 - 使用 NAVAJO 进行密码安全强度检测

## 关键字

数据加密 \ 密码强度 \ 输入检测 \ 安全检测 \ NAVAJO

## 需求场景

1. 对用户注册密码进行安全强度检测

## 参考链接

1. [GitHub - Navajo](#)

## 详细内容

```
NSString *password = @"abc123"
NJOPasswordValidator *validator = [NJOPasswordValidator standard
validator];

NSArray *failingRules = nil;
BOOL isValid = [validator validatePassword:password
                                    failingRules:&failingRules];

if (!isValid) {
    for (id <NJOPasswordRule> rule in failingRules) {
        NSLog(@"%@", [rule localizedErrorDescription]);
    }
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-03 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

数据存储 - 使用 SSZipArchive 实现文件的压缩和解压缩

## 关键字

数据存储 \ 文件 \ 压缩 \ 解压缩 \ zip \ unzip

## 需求场景

1. 需要对文件进行压缩和解压缩操作时

## 参考链接

1. [GitHub - SSZipArchive](#)

## 详细内容

1. 将 `SSZipArchive` 文件加入工程
2. 引入 `SSZipArchive.h` 头文件
3. 添加 `libz.dylib` 库
4. 用法```objectivec // Unzipping NSString zipPath = @"path\_to\_your\_zip\_file"; NSString destinationPath = @"path\_to\_the\_folder\_where\_you\_want\_it\_unzipped"; [SSZipArchive unzipFileAtPath:zipPath toDestination:destinationPath];

```
// Zipping NSString zippedPath = @"path_where_you_want_the_file_created";
NSArray inputPaths = [NSArray arrayWithObjects: [[NSBundle mainBundle]
pathForResource:@"photo1" ofType:@"jpg"], [[NSBundle mainBundle]
pathForResource:@"photo2" ofType:@"jpg"] nil];
[SSZipArchive
createZipFileAtPath:zippedPath withFilesAtPaths:inputPaths];
```

#### ##### Swift 使用实例

```
```swift
var zipPath = NSBundle.mainBundle().pathForResource("TestArchive",
", ofType: "zip")
var outputPath = NSSearchPathForDirectoriesInDomains(.DocumentDirectory,
.UserDomainMask, true)[0] as String
SSZipArchive.unzipFileAtPath(zipPath, toDestination: outputPath)
```

## Objective-C 使用实例

```
NSString *zipPath = [[NSBundle bundleForClass:[self class]] path
ForResource:@"TestArchive" ofType:@"zip"];
NSString *outputPath = [self _cachesPath:@"Regular"];
[SSZipArchive unzipFileAtPath:zipPath toDestination:outputPath d
elegate:self];
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-18 | Alfred Jiang | -  |

## 方案名称

数据存储 - 序列化对象

## 关键字

数据存储 \ 序列化对象 \ NSKeyedArchiver \ NSJSONSeriali \ 文件存储

## 需求场景

1. 需要文件存储对象数据时
2. 部分序列化对象需求

## 参考链接

1. [GitHub - NSSerialisationTests](#)

## 详细内容

通过NSKeyedArchiver和NSJSONSeriali的对比测试说明序列化对象的一些基本操作和用途

1. Model.h ````objectivec // // Model.h // Serialisation Test // // Created by Johnnie Walker on 09/05/2013. // Copyright (c) 2013 Random Sequence. All rights reserved. //

## import

```
@interface Model : NSObject
@property (nonatomic) NSRect rect;
@property (nonatomic) NSPoint point;
@property (nonatomic, copy) NSString *UUID;
```

- (id)model;
- (NSDictionary \*)dict;
- (id)modelWithDictionary:(NSDictionary \*)dictionary;

```
@end
```

```
2. Model.m
````objective-c
//
//  Model.m
//  Serialisation Test
//
//  Created by Johnnie Walker on 09/05/2013.
//  Copyright (c) 2013 Random Sequence. All rights reserved.
//

#import "Model.h"

NSString * const UUIDKey = @":UUID";
NSString * const PointKey = @":point";
NSString * const RectKey = @":rect";

@implementation Model

+ (id)model {
    Model *model = [Model new];
    model.UUID = [[NSUUID UUID] UUIDString];
    model.point = NSMakePoint(arc4random(), arc4random());
    model.rect = NSMakeRect(model.point.x, model.point.y, arc4random(), arc4random());
    return model;
}

- (void)encodeWithCoder:(NSCoder *)aCoder {
    if ([aCoder allowsKeyedCoding]) {
        [aCoder encodeObject:self.UUID forKey:UUIDKey];
        [aCoder encodePoint:self.point forKey:PointKey];
    }
}
```

```

        [aCoder encodeRect:self.rect forKey:RectKey];
    } else {
        [aCoder encodeObject:self.UUID];
        [aCoder encodePoint:self.point];
        [aCoder encodeRect:self.rect];
    }
}

- (id)initWithCoder:(NSCoder *)decoder {
    self = [super init];
    if (self) {
        if ([decoder allowsKeyedCoding]) {
            self.UUID = [decoder decodeObjectForKey:UUIDKey];
            self.point = [decoder decodePointForKey:PointKey];
            self.rect = [decoder decodeRectForKey:RectKey];
        } else {
            self.UUID = [decoder decodeObject];
            self.point = [decoder decodePoint];
            self.rect = [decoder decodeRect];
        }
    }
    return self;
}

- (NSDictionary *)dict {
    return @{
        UUIDKey: self.UUID,
        PointKey: NSStringFromPoint(self.point),
        RectKey: NSStringFromRect(self.rect)
    };
}

+ (id)modelWithDictionary:(NSDictionary *)dictionary {
    Model *model = [Model new];
    model.UUID = [dictionary objectForKey:UUIDKey];
    model.point = NSPointFromString([dictionary objectForKey:PointKey]);
    model.rect = NSRectFromString([dictionary objectForKey:RectKey]);
    return model;
}

```

```
}
```

```
@end
```

```
1. Serialisation_TestTests.m ````objectivec // // Serialisation_TestTests.m //  
Serialisation TestTests // // Created by Johnnie Walker on 09/05/2013. //  
Copyright (c) 2013 Random Sequence. All rights reserved. //
```

```
import "Serialisation_TestTests.h"
```

```
import "Model.h"
```

```
define ITERATIONS 100000
```

```
@interface Serialisation_TestTests () @property (nonatomic, strong) NSArray  
*models; @end
```

```
@implementation Serialisation_TestTests
```

- (void)setUp { NSMutableArray \*models = [NSMutableArray  
arrayWithCapacity:ITERATIONS]; for (NSInteger i=0; i<ITERATIONS; i++) {

```
[models addObject:[Model model]];
```

  
} self.models = models; }
- (void)tearDown { self.models = nil; }
- (void)testNSCoding { NSDate *startDate* = [*NSDate date*]; NSMutableArray  
array = [NSMutableArray arrayWithCapacity:ITERATIONS]; for (Model \*model  
in self.models) {

```
[array addObject:[NSArchiver archivedDataWithRootObject:mo  
del]];
```

```
 } NSMutableArray models1 = [NSMutableArray
arrayWithCapacity:ITERATIONS]; for (NSData data in array) {

    [models1 addObject:[NSUnarchiver unarchiveObjectWithData:data]];
}

NSDate *endDate = [NSDate date];

NSLog(@"NSArchiver models: %li duration: %f", (unsigned long)[models1
count], [endDate timeIntervalSinceDate:startDate]); }

• (void)testNSKeyedCoding { NSDate startDate = [NSDate date];
NSMutableArray array = [NSMutableArray arrayWithCapacity:ITERATIONS];
for (Model *model in self.models) {

    [array addObject:[NSKeyedArchiver archivedDataWithRootObject:model]];
}

NSMutableArray models1 = [NSMutableArray
arrayWithCapacity:ITERATIONS]; for (NSData data in array) {

    [models1 addObject:[NSKeyedUnarchiver unarchiveObjectWithData:data]];
}

NSDate *endDate = [NSDate date];

NSLog(@"NSKeyedArchiver models: %li duration: %f", (unsigned long)
[models1 count], [endDate timeIntervalSinceDate:startDate]); }

• (void)testPropertyListSerialisation { NSDate startDate = [NSDate date];
NSMutableArray array = [NSMutableArray arrayWithCapacity:ITERATIONS];
for (Model *model in self.models) {

    [array addObject:[NSPropertyListSerialization dataWithPropertyList:[model dict] format:NSPropertyListBinaryFormat_v1_0
options:0 error:nil]];
}
```

```
} NSMutableArray models1 = [NSMutableArray
arrayWithCapacity:ITERATIONS]; for (NSData data in array) {

    [models1 addObject:
        [Model modelWithDictionary:
            [NSPropertyListSerialization propertyListWithData:data
                options:NSPropertyListImmutable
                format:NULL
                error:nil]]];
}

NSDate *endDate = [NSDate date];

NSLog(@"%@", NSPropertyListSerialization models: %li duration: %f", (unsigned
long)[models1 count], [endDate timeIntervalSinceDate:startDate]); }

• (void)testJSONSerialisation { NSDate startDate = [NSDate date];
NSMutableArray array = [NSMutableArray arrayWithCapacity:ITERATIONS];
for (Model *model in self.models) {

    [array addObject:[NSJSONSerialization dataWithJSONObject:[
        model dict] options:0 error:nil]];
}

NSMutableArray models1 = [NSMutableArray
arrayWithCapacity:ITERATIONS]; for (NSData data in array) {

    [models1 addObject:[Model modelWithDictionary:[NSJSONSerialia
zation JSONObjectWithData:data options:0 error:nil]]]];
}

NSDate *endDate = [NSDate date];

NSLog(@"%@", NSJSONSerialization models: %li duration: %f", (unsigned long)
[models1 count], [endDate timeIntervalSinceDate:startDate]); }

@end
```

#### 4. 执行测试结果

```
```objective-c
Test Case '-[Serialisation_TestTests testJSONSerialisation]' started.
2015-03-02 15:02:50.335 Serialisation Test[37221:303] NSJSONSerialisation models: 100000 duration: 2.247854
Test Case '-[Serialisation_TestTests testJSONSerialisation]' passed (2.516 seconds).
Test Case '-[Serialisation_TestTests testNSCoding]' started.
2015-03-02 15:02:53.854 Serialisation Test[37221:303] NSArchiver models: 100000 duration: 2.789287
Test Case '-[Serialisation_TestTests testNSCoding]' passed (2.967 seconds).
Test Case '-[Serialisation_TestTests testNSKeyedCoding]' started
.
2015-03-02 15:03:01.146 Serialisation Test[37221:303] NSKeyedArchiver models: 100000 duration: 6.989706
Test Case '-[Serialisation_TestTests testNSKeyedCoding]' passed (7.217 seconds).
Test Case '-[Serialisation_TestTests testPropertyListSerialization]' started.
2015-03-02 15:03:04.015 Serialisation Test[37221:303] NSPropertyListSerialization models: 100000 duration: 2.544815
Test Case '-[Serialisation_TestTests testPropertyListSerialization]' passed (2.787 seconds).
Test Suite 'Serialisation_TestTests' finished at 2015-03-02 07:03:04 +0000.
Executed 4 tests, with 0 failures (0 unexpected) in 15.488 (16.354) seconds
Test Suite '/Users/ajiang048/Library/Developer/Xcode/DerivedData/Serialisation_Test-betqtfwfsrkofaxdppxjckonkic/Build/Products/Debug/Serialisation TestTests.octest(Tests)' finished at 2015-03-02 07:03:04 +0000.
Executed 4 tests, with 0 failures (0 unexpected) in 15.488 (16.354) seconds
Test Suite 'All tests' finished at 2015-03-02 07:03:04 +0000.
Executed 4 tests, with 0 failures (0 unexpected) in 15.488 (16.356) seconds
```

## 效果图

(无)

## 备注

NSJSONSerialization 比 NSKeyedArchiver 在速度更快，而且序列化之后的体积更小，推荐优先使用 NSJSONSerialization 方式进行序列化操作

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-05-12 | Alfred Jiang | -  |

## 方案名称

数据存储 - 文件(目录)操作笔记

## 关键字

数据存储 \ 文件操作 \ 文件夹 \ 文件目录操作 \ 笔记

## 需求场景

1. 进行文件操作
2. 管理数据文件

## 参考链接

1. [Sina - NSJSONSerialization-JSON数据与NSDictionary和NSArray之间的转化](#)

## 详细内容

1. 创建文件目录

```
- (NSString *)dirFullPath:(NSString *)dirName
{
    NSArray *path = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);
    NSString *documentsDirectory = [path objectAtIndex:0];
    NSString *dirPath = [documentsDirectory stringByAppendingPathComponent:dirName];

    BOOL isDirectory = NO;

    if ([[NSFileManager defaultManager] fileExistsAtPath:dirPath
isDirectory:&isDirectory]) {

        if (isDirectory) {

            return dirPath;
        }
    }

    [[NSFileManager defaultManager] createDirectoryAtPath:dirPath
withIntermediateDirectories:YES attributes:nil error:NULL];

    return dirPath;
}
```

## 2. 读取文件目录下全部文件

```
NSArray *directoryContents = [[NSFileManager defaultManager] subpathsAtPath:dirFullPath];
```

## 3. 向文件目录添加文件，超过指定最大个数 (**MAX\_FILES\_COUNT**) 移除多余文件

```
NSArray *directoryContents = [[NSFileManager defaultManager] subpathsAtPath:dirFullPath];

NSArray *sortedList = [directoryContents sortedArrayUsingComparator:^NSComparisonResult(id _Nonnull obj1, id _Nonnull obj2) {

    return [obj2 compare:obj1];
}];

if ([sortedList count] > MAX_FILES_COUNT) {

    NSArray *subList = [sortedList subarrayWithRange:NSMakeRange(MAX_FILES_COUNT, ([sortedList count] - MAX_FILES_COUNT))];

    [subList enumerateObjectsUsingBlock:^(NSString *obj, NSUInteger idx, BOOL * _Nonnull stop) {

        NSLog(@"%@", obj);
        [[NSFileManager defaultManager] removeItemAtPath:[[self dirPathForURL:URLString] stringByAppendingPathComponent:obj] error:&nil];
    }];
}
```

## 效果图

(无)

## 备注

- 语法 - NSString \ NSData \ NSArray \ NSDictionary 格式转换 ( NSArray / NSDictionary <=> NSData <=> NSString )
- NSString - 筛选出 NSString 中特定字符串

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-04-23 | Alfred Jiang | -  |

## 方案名称

数据库 - 使用 SQLCipher 进行数据库加密存储

## 关键字

数据库 \ SQLCipher \ 加密

## 需求场景

1. 需要对存储于数据库中的数据进行加密时

## 参考链接

1. CSDN - sqlite 数据库加密 (SQLCipher)
2. CSDN - ios 中的 SQL 数据库文件加密 (使用 sqlcipher)
3. SQLCipher - Adding SQLCipher to Xcode Projects(推荐)

## 详细内容

### 1. 添加

可参考 [SQLCipher - Adding SQLCipher to Xcode Projects](#)

### 2. 使用

参考以下示例

```
//  
//  AppDelegate.m
```

```
// SecureLoginDelegate
//
// Created by Billy Gray on 10/19/14.
// Copyright (c) 2014 Zetetic. All rights reserved.
//

#import "AppDelegate.h"
#import <sqlite3.h>

@interface AppDelegate ()
@property (nonatomic) BOOL isLoggedInViewControllerDisplayed;
@property (readonly) NSURL *databaseURL;
@property (readonly) BOOL databaseExists;
@end

@implementation AppDelegate
@dynamic databaseURL;
@dynamic databaseExists;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Set up the window with loginViewController as the rootView
    // controller for now
    // to avoid showing app view on launch on iOS 8
    [[self window] setRootViewController:self.loginViewController];
    [[self window] makeKeyAndVisible];
    self.isLoggedInViewControllerDisplayed = YES;

    // Set up a SQLCipher database connection:
    sqlite3 *db;
    if (sqlite3_open([[self.databaseURL path] UTF8String], &db) == SQLITE_OK) {
        const char* key = @"StrongPassword" UTF8String];
        sqlite3_key(db, key, (int)strlen(key));
        if (sqlite3_exec(db, (const char*) "SELECT count(*) FROM
        sqlite_master;", NULL, NULL, NULL) == SQLITE_OK) {
            NSLog(@"Password is correct, or a new database has b
            een initialized");
        } else {
    
```

```

        NSLog(@"Incorrect password!");
    }
    sqlite3_close(db);
}
return YES;
}

- (NSURL *)databaseURL {
    NSArray *URLs = [[NSFileManager defaultManager] URLsForDirectory:NSDocumentDirectory inDomains:NSUserDomainMask];
    NSURL *directoryURL = [URLs firstObject];
    NSURL *databaseURL = [directoryURL URLByAppendingPathComponent:@"secure.db"];
    return databaseURL;
}

- (BOOL)databaseExists {
    BOOL exists = NO;
    NSError *error = nil;
    exists = [[self databaseURL] checkResourceIsReachableAndReturnError:&error];
    if (exists == NO && error != nil) {
        NSLog(@"Error checking availability of database file: %@", error);
    }
    return exists;
}

@end

```

### 3. 注意事项

若出现以下错误提示：

- No architectures to compile for (ARCHS=armv6,armv7,  
VALID\_ARCHS=armv7 armv7s...)

Bulid Settings -> Architectures和Valid Architectures里面都改成一样（例如：都填写 armv6 armv7）

- warning: implicit declaration of function 'sqlite3\_key' is invalid in C99

Build Settings -> C Language Dialect 改为 : C89[-std-c89] 就可以，即使用c89标准

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-05-10 | Alfred Jiang | -  |

## 方案名称

文档 - iOS 人机交互指南

## 关键字

文档 \ 人机交互指南 \ 人机界面指南

## 需求场景

1. 了解 iOS 设计基础

## 参考链接

- Apple documentation - iOS Human Interface Guidelines
- 腾讯ISUX - [ISUX译]iOS 9人机界面指南(一)：UI设计基础
- 腾讯ISUX - [ISUX译]iOS 9人机界面指南(二)：设计策略
- 腾讯ISUX - [ISUX译]iOS 9人机界面指南(三)：iOS 技术 (上)
- 腾讯ISUX - [ISUX译]iOS 9人机界面指南(三)：iOS 技术 (下)
- 腾讯ISUX - [ISUX译]iOS 9人机界面指南(四)：UI元素
- 腾讯ISUX - [ISUX译]iOS 9人机界面指南(五)：图标与图形设计
- 知乎专栏 - 《iOS 人机界面准则》中文版

## 详细内容

(见参考链接)

## 效果图

(无)

## 备注

- 界面设计 - 相关资料收集

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-08-19 | Alfred Jiang | -  |
| 2  | 2015-12-20 | Alfred Jiang | -  |

## 方案名称

文档 - iOS 项目的目录结构

## 关键字

文档 \ Xcode \ 工程 \ 目录 \ 项目目录 \ 结构

## 需求场景

1. 创建 iOS 项目工程，组织工程目录

## 参考链接

1. CSDN - iOS 项目的目录结构能看出你的开发经验
2. Limboy - iOS 项目的目录结构和开发流程

## 详细内容

iOS 工程目录的结构根据工程规模大小大致可分为以下两类

1. 主目录按照业务分类，内目录按照模块分类(主目录按照MVC架构分类，内部根据项目模块分类)

优点：能比较快定位对应的业务。

缺点：模块相关类太过分散，需要来回切换文件，不方便开发。

适合中小规模工程，各模块耦合度高，同时参与开发人员较少时适用

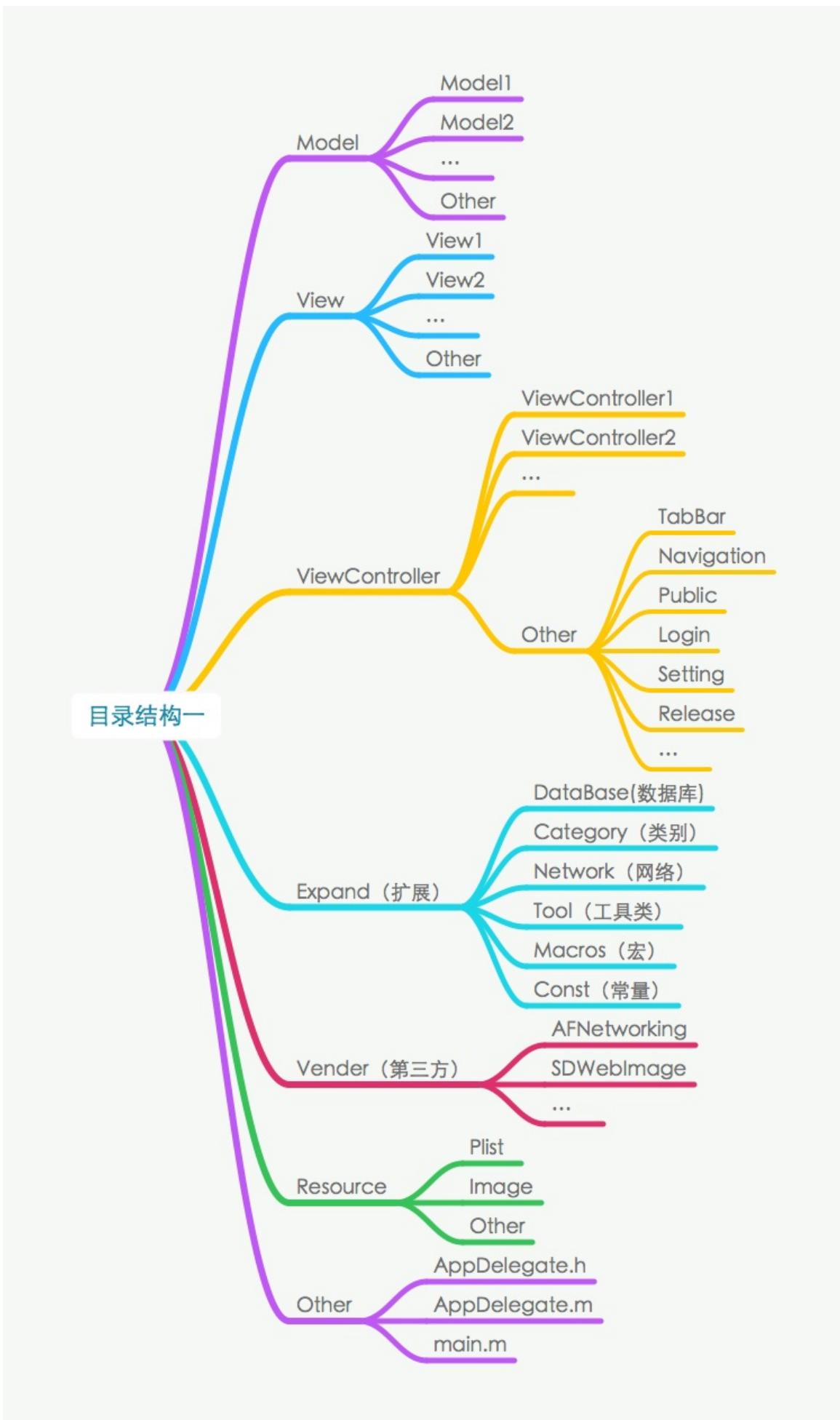
## 2. 主目录按照模块分类，内目录按照业务分类

优点：对某一模块用到的类集中化，方便管理与开发，对新手来说，不用来回在其他文件中切换、寻找对应的模块类。

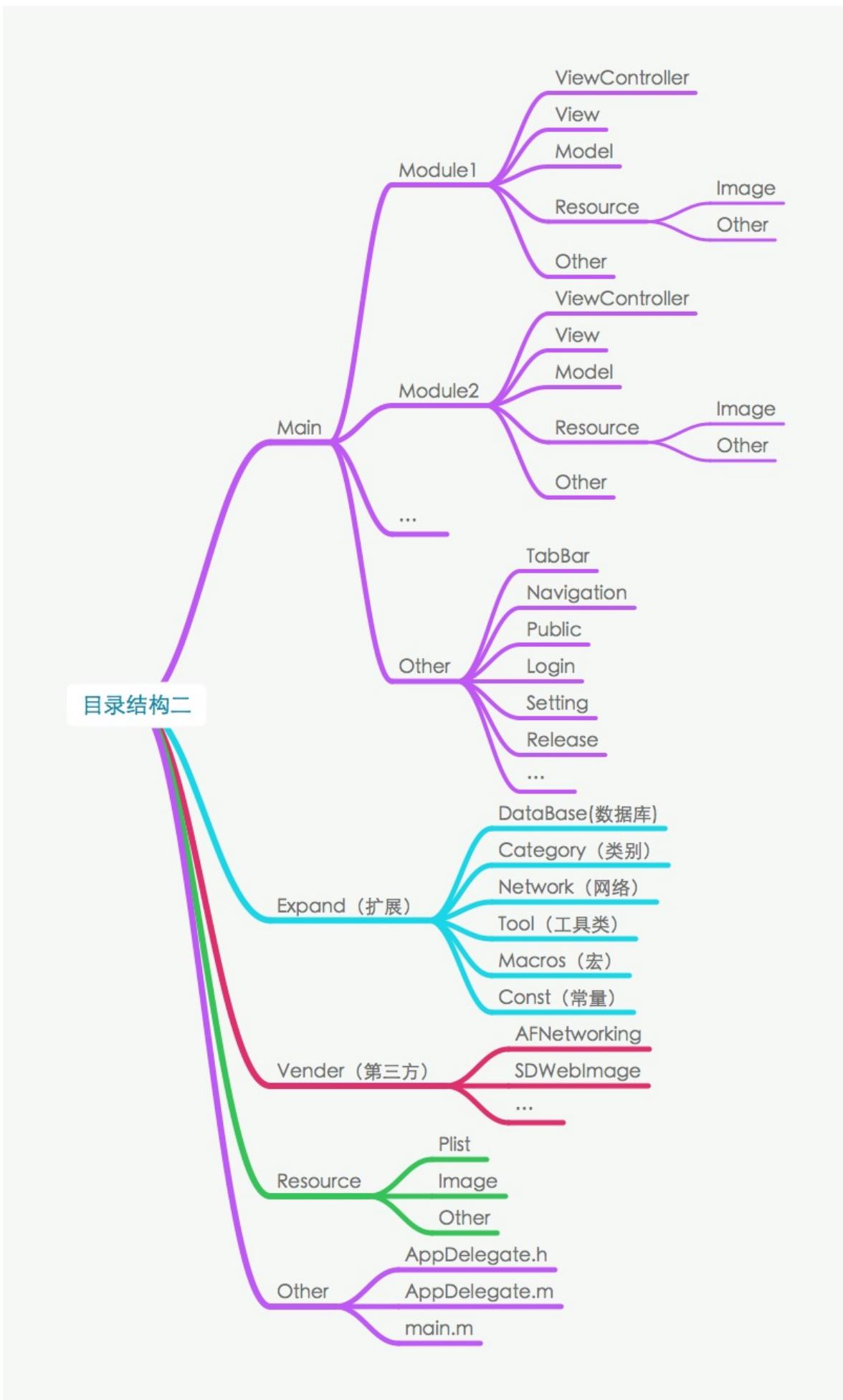
缺点：当几个模块公用一些类时，不太好归类。

适合较大规模工程，各模块耦合度低，同时参与开发人员较多时适用

### 效果图







## 备注

1. 建议工程目录结构对应物理文件夹结构，即目录分组则创建对应文件夹
2. 对于.h.m.xib文件或.swift.xib同名文件放在同一组，同时对应存放在同一文件夹中
3. 对于**Resource**中图片文件建议物理分组（文件夹分组），同时按照对应模块进行前缀命名，对于公共图片考虑存放在单独的公共文件夹中

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-08-18 | Alfred Jiang | -  |
| 2  | 2015-12-20 | Alfred Jiang | -  |
| 3  | 2016-01-26 | Alfred Jiang | -  |

## 方案名称

文档 - 苹果审核未通过问题与解决方案参考

## 关键字

文档 \ 审核 \ App Store

## 需求场景

- 提交到 App Store 的应用被拒绝时参考

## 参考链接

- CocoaChina - 苹果App Store审核指南中文翻译（更新）
- Apple documentation - App Store Review Guidelines
- CocoaChina - 总结：2015这一年App Store审核指南都有哪些变化

## 详细内容

### 前言

感谢您付出宝贵的才华与时间来开发iOS应用程序。从职业与报酬的角度而言，这对于成千上万的开发员来说一直都是一项值得投入的事业，我们希望帮助您加入这个成功的组织。我们发布了《App Store审核指南》（App Store Review Guidelines），希望通过它帮您避开开发应用程序过程中的一些问题，并帮你在提交应用时加快审核流程。

我们将应用程序（Apps）视为与书籍或歌曲不同的产品，我们并不存储它们。如果您意欲批评宗教，那就去写本书。如果您想要描述性，那就写本书或写首歌，或者可以创建一个医疗程序。这么做可能会比较复杂，但是我们不允许在应用程序商店（App Store）出现某种禁止内容。这会让您认识到我们秉持的更为深远的目的：

我们有很多可供儿童可以下载的应用程序。家长的监护可以很好地保护孩子，但是您需要做您应该做的那一部分。因此，您要了解我们时刻在留心着您的孩子。

App Store中有数百万的应用。如果您的应用程序没有什么有益的用途，不是独一无二的或者不能提供持续性的娱乐功能，那它可能不会被我方接受。

如果您的应用程序看上去像是那种只花了几分钟简单拼凑出来的产品，或者只是想在我们的商店中抓住朋友的眼球，请提前做好被拒的准备。我们有很多态度严谨的开发者，不希望他们的高品质应用程序充斥在一些业余作品之中。

我们将拒绝任何包含越界内容或行为的应用程序。您可能会问道，具体限制是什么？最高法院的法官曾有言："它出现时我自然心中有数。"当您越过这一范围时，我们认为您也会有自知之明。

如果您的应用程序被拒，我们设立了一个审查委员会供您上诉。如果您去媒体抨击我们，肯定对您于事无补。

如果您试着作弊（比如在审核流程中作假，窃取用户数据，抄袭其他开发者作品，或者操作应用评分），我们将会移除您的应用程序，并且将您从开发者计划中除名。

这是一个动态文档，新提交的应用程序会导致新的问题产生，并可能随时产生新的规则。或许您的应用程序会触及到这一点。

最后要说明的是，我们非常珍惜这个平台，并且向您的作品表示敬意。我们确实在尝试尽力创建全球最佳平台，以便让您展示才华，同时获得相应的报酬。如果这读上去让您感觉我们的控制欲过强，那是因为我们曾向用户承诺保证，我们将利用我们的产品让他们获得高品质体验。

## 目录

### 1. 条款与条件

### 2. 功能

3. 元数据

4. 位置

5. 推送通知

6. 游戏中心

7. 广告

8. 内容和知识产权

9. 媒体内容

10. 用户界面

11. 购买与货币

12. 抓取与聚合

13. 设备损害

14. 人身攻击

15. 暴力

16. 令人反感的内容

17. 隐私

18. 色情

19. 宗教、文化与种族

20. 竞赛、赌博、彩票和抽奖

21. 慈善与援助

22. 法律要件

23.Wallet

24. 儿童类别

25. 扩展

26.HomeKit

27.HealthKit

28.TestFlight

29.Apple Pay

## 1. 条款和条件

1.1 为App Store开发程序，开发者必须遵守Program License Agreement(PLA)、人机交互指南(HIG)以及开发者和苹果签订的任何其他协议和合同。以下规则和例证旨在帮助开发者的程序能获得App Store的认可，而不是修改或删除任何其他协议中的条款。

## 2. 功能

2.1 崩溃的程序将会被拒绝。

2.2 存在错误的程序将会被拒绝。

2.3 跟开发者宣传不符的程序将会被拒绝。

2.4 无应用文档或隐藏功能与描述不符的程序将会被拒绝。

2.5 使用非公开API的程序将会被拒绝。

2.6 在指定容器范围外读写数据的程序将会被拒绝。

2.7 以任何方式或形式下载代码的程序将会被拒绝。

2.8 安装或运行其他可执行代码的程序将会被拒绝。

- 2.9 Demo版、trial版和test版的程序将会被拒绝。 Beta版应用程序仅能通过Test Flight提交，并且必须遵守相关指南。
- 2.10 iPhone程序必须不经修改就能以iPhone分辨率和2倍 iPhone 3GS的分辨率在iPad上运行。
- 2.11 与App Store已有程序重复的应用可能会被拒绝，特别是数量很多的情况下，比如手电筒应用和爱经应用。
- 2.12 没有显著用途、不独特的应用程序或者与网站简单捆绑的应用有可能被拒；不提供任何持久娱乐价值的程序可能会被拒绝。
- 2.13 内容主要是营销材料或广告的程序将会被拒绝。
- 2.14 包含欺骗或虚假功能，却有没有标明的应用程序将会被拒绝。
- 2.15 大于100MB无法通过蜂窝网络下载的应用（App Store会自动禁止）。
- 2.16 多任务程序使用后台服务仅限于几种目的：VoIP、音频播放、地理位置、完成任务以及本地提醒等。
- 2.17 应用程序只允许使用iOS WebKit框架和WebKit Javascript浏览web内容。
- 2.18 鼓励酗酒或使用违禁药物，或引诱青少年饮酒或吸烟的程序将会被拒绝。
- 2.19 提供错误的系统诊断或不精确的设备数据的应用将会被拒绝。
- 2.20 向App Store上传大量相似版本程序的开发者将会从iOS开发者计划中除名。
- 2.21 简单一首歌曲或者一部影片应用要提交到iTunes store，书籍类应用应该提交到iBookstore。
- 2.22 武断地根据环境（如定位或者运营商）限制用户使用的应用会被拒。
- 2.23 应用必须遵守iOS数据储存指导方针（iOS Data Storage Guidelines），否则应用将被拒。
- 2.24 存放在Newsstand的应用必须遵守开发者项目许可协议（Program License Agreement）的表1、表2以及表3，否则应用将会被拒。

2.25 使用用户将其与App Store混淆，或者基于购买或者促销的目的而展示其他应用的应用将会被拒绝。

2.26 只有当app是出于特殊审核需要（比如健康管理、航空以及无障碍需求等）或为特殊群体用户提供具有重大意义的附加值时，才可以展示和推荐自身以外的其他应用程序，否则应用程序将会被拒绝。

2.27 如果App的核心功能不能使用Siri遥控器，那么App将会被拒绝。不过应用程序可以提供与游戏控制器或者其他周边设备相连接的增强功能。（2015.10 新增）

### 3. 元数据（名称、描述、评级、排名等）

3.1 应用或者元数据中提到其他任何移动平台将会被拒。

3.2 带有占位符文本的程序将会被拒绝。

3.3 应用程序的名称、描述、截图或者预览与应用的内容和功能不相关将会被拒绝。  
(旧版：3.3 描述中有与程序内容和功能不相关的信息的应用将会被拒绝。)

3.4 为了不混淆用户，iTunes Connect中的应用名称应该和展示在设备上的应用名称一致。

3.5 不同尺寸的app icon要一致，否则会造成混淆。

3.6 应用程序的icon、截图、预览以及位于Apple TV主屏幕顶部展区的Apple TV app的图片展示不符合4+年龄评级的程序将会被拒绝。（2015.10 修正）

(旧版：应用程序的icon、截图以及预览不符合4+年龄评级的程序将会被拒绝。)

3.7 目录与类型不适合于程序内容的程序将会被拒绝。

3.8 开发者有责任为其程序指定适合的评级。不相称的评级可能会由苹果公司修改。

3.9 开发者有责任为其程序指定恰当的关键字。不恰当的关键词可能会被苹果公司修改/删除。

3.10 试图通过伪造评论或者付费评论的方式在AppStore中操纵或者其欺骗用户评论（或者采用其他不正当方式）以提升排名的开发者将会被苹果从iOS开发者计划中除名。

3.11 在安装或打开应用之前，推荐用户重启iOS设备的应用将会被拒。

3.12 提交审核的应用程序应包含能正常运行的URL，比如支持服务URL和隐私政策URL。

3.13 应用程序的截图、预览或者营销文本没有清晰地指出附加内容或项目需要额外单独购买（比如使用IAP）将会被拒绝。

3.14 App预览仅能使用从应用程序捕获的视频屏幕、旁白、文本以及design overlays，否则应用程序将会被拒绝。

3.15 添加App预览的应用程序，未经许可展示真人个人信息将会被拒绝。

3.16 App预览仅能使用在所有选定地区内经过授权许可、用于此目的的音乐。

3.17 App预览和截图包含未经授权的通过App（如音乐、视频以及以及相关封面艺术设计）播放的内容或流媒体将会被拒绝。（2015.10 修正）

（旧版：App预览包含未经授权的通过app播放的内容（比如iTunes playlist和YouTube流媒体）的应用将会被拒绝。）

#### 4. 位置

4.1 在收集、传输或使用位置数据之前未通知并获得用户同意的程序将会被拒绝。

4.2 将基于位置的API用于车辆、飞机或其他设备的自动控制或自主控制的应用程序将会被拒绝。

4.3 将基于位置的API用于应急服务的应用程序将会被拒绝。

（旧版：4.3 将基于位置的API用于调度、车队管理或应急服务的程序将会被拒绝。）

4.4 当与提供的功能或服务密切相关，或者为支持经过授权的广告时，应用程序才可以使用位置数据。

#### 5. 推送通知

5.1 不使用苹果推送通知（APN）应用接口提供推送通知的程序将会被拒绝。

5.2 未从苹果获得Push Application ID便擅自使用APN服务的程序将会被拒绝。

5.3 在首次推送消息或者要求运行推送通知之前未获得用户许可的应用将会被拒绝。

- 5.4 使用推送通知发送敏感个人信息或机密信息的程序将会被拒绝。
- 5.5 使用推送通知发送非请求消息，或用于钓鱼或群发垃圾信息用途的程序将会被拒绝。
- 5.6 应用程序不可使用推送通知发送广告、促销或任何类型的直销信息。
- 5.7 应用程序不能向使用推送通知服务的用户收取费用。
- 5.8 使用推送通知会过多利用APN服务的网络流量或带宽或给设备带来过度负担的程序将会被拒绝。
- 5.9 如果应用程序传送病毒、文件、计算机代码或程序，并且对APN服务的正常运行造成损害或中断，那么该程序将会被拒绝。

## 6. 游戏中心

- 6.1 向终端用户或任意第三方显示玩家ID的程序将会被拒绝。
- 6.2 将玩家ID用于任何未经游戏中心条款批准用途的程序将会被拒绝。
- 6.3 试图进行反向搜索、跟踪、关联、挖掘、获得或利用玩家ID、别名或通过游戏中心获得其他信息的开发者将会iOS开发者计划除名。
- 6.4 游戏中心信息（例如排行榜分数），只能用于游戏中心批准的应用程序中。
- 6.5 利用游戏中心服务发送非请求信息，或用于钓鱼或群发垃圾邮件的程序将会被拒绝。
- 6.6 过多使用游戏中心网络流量或带宽的应用程序将会被拒绝。
- 6.7 如果程序能够传送病毒、文件、计算机代码或程序，并且对游戏中心服务的正常运行造成损害或中断，该程序将会被拒绝。

## 7. 广告

- 7.1 人工刷广告浏览量或者广告点击率的应用程序将会被拒绝。
- 7.2 包含空iAd广告的应用程序将会被拒绝。

7.3 主要设计目的在于显示广告的应用程序将会被拒绝。

## 8. 内容与知识产权

8.1 应用程序必须遵守"Guidelines for Using Apple Trademarks and Copyrights"和"Apple Trademark List"中说明的所有条款与条件。

8.2 任何误导和暗示苹果公司是该应用程序来源或提供商，或者苹果公司以任何形式表示认可其质量或功能的应用程序将会被拒绝。

8.3 与目前已有苹果产品或者广告主题外观相似或混淆的应用程序将会被拒绝。

8.4 在应用程序名称中将苹果产品名拼错的应用程序（例如，GPS for Iphone，iTunz）将会被拒绝。

8.5 应用程序不得使用受保护的第三方材料（比如商标、版权以及专利），不能违反第三方使用条款。必须提供使用这些材料的授权许可。

8.6 若无明确授权许可，从第三方来源处（比如YouTube、SoundCloud以及Vimeo等）下载音乐或者视频内容的应用程序将会被拒绝。（2015.3 新增）

## 9. 媒体内容

9.1 不使用媒体播放器框架（MediaPlayer Framework）获取音乐库中媒体内容的应用程序将会被拒绝。

9.2 用户界面模仿任何iPod或者iTunes界面的应用程序将会被拒绝。

9.3 通过蜂窝网络传输的音频流内容每5分钟不得超过5MB。

9.4 通过蜂窝网络传输超过10分钟的视频流内容必须使用HTTP Live Streaming协议，并且要包含一个基线为192kbps或者更低的HTTP实时流。（2015.3 修正）

## 10. 用户界面

10.1 应用程序必须遵守苹果 iOS Human Interface Guidelines 中所有的条款和条件：iOS Human Interface Guidelines，OS X Human Interface Guidelines，Apple TV Human Interface Guidelines以及Apple Watch Human Interface Guidelines. (2015.10 修正)

(旧版：应用程序必须遵守苹果的《iOS Human Interface Guidelines》中所有的条款和条件。)

**10.2 外观与iOS或Watch OS设备自带应用（比如App Store、iTunes Store和iBookstore）相似的应用程序将会被拒绝。（2015.4 修正）**

**10.3 未能按苹果《iOS Human Interface Guidelines》描述正确使用系统提供的项目（比如按钮、图标）的应用将会被拒绝。**

**10.4 创建桌面/主屏幕环境或者模拟multi-App插件体验的应用程序将会被拒绝。**

**10.5 修改音量大小和铃声/静音等标准开关功能的应用程序将会被拒绝。**

**10.6 苹果和我们的客户高度推崇简单、精致、富有创造性以及经过精心设计的界面。虽然需要付出更多，但却非常值得。苹果设立了很高的门槛。如果你的用户界面太过复杂或者水准不高，可能会被拒绝。**

**10.7 主要功能为报时的Watch app将会被拒。（2015.4 新增）**

## 11. 购买与货币流通

**11.1 使用App Store以外的渠道解锁或开启附加属性和功能的应用程序将会被拒绝。**

**11.2 使用应用内支付系统（IAP）以外的系统购买内容、功能或服务的应用软件将会被拒绝。**

**11.3 使用IAP购买实物商品或者用于该软件之外的商品和服务的应用软件将会被拒绝。**

**11.4 使用IAP购买积分（信用点）或者其他货币必须在本应用中消费。**

**11.5 使用IAP购买已过期积分或其他货币的应用软件将会被拒绝。**

**11.6 使用IAP订阅的内容至少要持续7天，而且允许在用户的其他iOS设备间共享。**

**11.7 使用IAP购买项目的应用程序必须指派正确的购买类型。**

**11.8 使用IAP购买使用iOS、watchOS以及tvOS内置功能（如照相机，陀螺仪）或者苹果品牌周边产品（比如苹果键盘、苹果电容笔）的应用程序将会被拒绝。（2015.10**

修正)

旧版：使用IAP购买iOS内置功能（如照相机，陀螺仪）的应用程序将会被拒绝。

**11.9** 含有超过限定时间的内容或服务的应用程序将会被拒绝，除经特定批准的内容（比如电影、电视节目音乐以及书籍）。

**11.10** 保险类应用程序必须免费，要遵守发布地区的法律，并且不能使用IAP。

**11.11** 一般而言，你的应用程序越贵，我们的评审会越深入。

**11.12** 提供订阅功能的应用必须使用IAP，苹果将会按照 *Developer Program License Agreement* 中的约定与开发者按30/70比例分成。

**11.13** 在应用内使用跳转至外部购买或订阅链接的应用将会被拒，比如"buy"按钮跳转至一个购买电子书的web页面。

**11.14** 只要应用内没有跳转至外部购买、订阅的按钮或链接，苹果允许这些应用读取或展示经批准的、在应用外购买或订阅内容(特别是杂志、报纸、书籍、音频、音乐、视频以及云存储内容)。苹果只能通过应用程序内的购买获得一部分收益。

**11.15** 应用程序可以只使用自动更新订阅期刊(报纸、杂志)、商业应用程序(企业类、效率类、专业创意类以及云存储类)和媒体类应用程序(视频、音频、声音)，否则应用程序将被拒绝。

**11.16** 当与特定的经过审核的实体产品（比如玩具）结合使用时，应用程序可以使用获得批准的附加特性和功能，只要附加功能完全依赖于该硬件产品（比如一款用于控制望远镜的应用程序）或者也可以在不使用实物产品的情况下使用应用程序，比如成就奖励或者使用IAP。

**11.17** 如果应用功能遵照各州和联邦法律，那么应用可以用来促进被认可的虚拟货币的流通。

## 12. 抓取和聚合

**12.1** 从苹果网站（例如apple.com、iTunes Store、App Store、iTunes Connect以及Apple Developer Programs等）抓取任何信息或者使用苹果网站内容和服务进行排名的应用程序将会被拒绝。

**12.2** 应用软件可以使用获得批准的苹果RSS feeds，例如iTunes Store RSS feed

ds。

12.3 只是简单的网页剪切、内容整合或者收集链接的应用程序可能会被拒绝。

### 13. 损害设备

13.1 忿恿用户以可能造成损害的方式使用苹果设备的应用软件将会被拒绝。

13.2 快速耗光设备电量或产生过多热量的应用软件将会被拒绝。

13.3 能导致用户人身伤害的app将会被拒绝。

### 14. 人身攻击

14.1 涉及诽谤、人身攻击性质以及内容狭隘卑鄙的应用软件或者打击特定个人或组织的应用软件将会被拒绝。

14.2 职业政治讽刺家和幽默作家不受这一条款约束。

14.3 展示用户创作内容（UGC）的应用程序必须提供一个过滤不良资讯的方法，一个用户可以标记侵犯性内容的机制，以及可以阻止辱骂用户的能力。

### 15. 暴力

15.1 应用程序中出现人或动物被杀、致残以及枪击、刺伤、拷打等受伤情形的真实画面将会被拒绝。

15.2 出现描绘暴力或虐待儿童等内容的应用程序将会被拒绝。

15.3 游戏中出现的“敌人”不可指向一个特定种族、文化、一个真实存在的政府、企业或者其他任何现实中的实体。

15.4 对武器进行真实描述以怂恿非法使用或滥用这些武器的应用程序将会被拒绝。

15.5 包含俄罗斯轮盘赌博内容的游戏将会被拒。

### 16. 令人反感的内容

16.1 应用程序中出现过于令人反感或者低俗的内容将会被拒绝。

16.2 在设计上激怒用户或令人感到厌恶的应用程序将会被拒绝。

## 17. 隐私

17.1 在未经用户事先许可，或未告知用户如何使用信息以及在何处使用信息的情况下，应用程序不能传输用户数据。

17.2 要求用户共享电子邮箱地址和出生日期等私人信息才可使用其功能的应用程序将会被拒绝。

17.3 仅出于遵守适用的儿童隐私法规的目的，应用程序可以要求用户的出生日期（或者使用其他年龄评级机制），但是必须包括一些有用的功能或者娱乐价值，不管用户年龄大小。

17.4 收集、传输以及分享未成年用户个人信息（比如名字、地址、邮件、位置、照片、视频、绘画、聊天信息以及其他个人数据，或者与以上所述相关的永久性标示符）的应用程序必须遵守应用儿童隐私法规，并且必须包含隐私条款。

17.5 包含账号注册或者访问用户现有账号的应用程序必须包含隐私策略，否则将会被拒绝。

## 18. 色情

18.1 含有色情素材，也就是《韦氏词典》中定义的“旨在激发情欲，对性器官或性行为的明确描述或展示，而无关美学或情绪感受”的程序将会被拒绝。

18.2 包含用户频繁提供的色情内容的应用程序（比如以前的“Chat Roulette”程序）将会被拒绝。

## 19. 宗教，文化与种族

19.1 涉及宗教、文化或种族群体的引用或评论包含诽谤性、攻击性或狭隘内容，或会使特定群体遭受伤害或暴力的应用程序将会被拒绝。

19.2 程序可以包含或引用宗教经文，程序所提供的引用或翻译必须准确且不会引起误导。评论应该有教育意义，可以令人开阔眼界，而不应有煽动性。

## 20. 竞赛、赌博、彩票以及抽奖

20.1 彩票抽奖和竞赛必须由应用程序的开发者或者app所属公司发起。

20.2 应用程序必须展示彩票抽奖和竞赛的正式规则，并声明苹果不是发起者，也没有以任何方式参与活动。

20.3 开发者运营一款具有抽奖性质的应用必须经过法律允许，并且抽奖应用必须具备以下特征：报酬、运气以及奖品。

20.4 允许用户在应用中直接购买彩票或彩券的应用将会被拒。

20.5 提供真钱游戏（比如体育博彩、扑克牌、赌场游戏、赛马以及彩票）的应用程序必须有应用程序适用地区当地必要的许可和允许，必须限制在这些区域，必须可以从App Store免费下载。

20.6 使用IAP购买信誉或者货币，且结合真钱游戏的应用将会被拒绝。

## 21. 慈善与援助

21.1 包含可以向已认证的慈善组织捐赠功能的应用程序必须是免费的。

21.2 捐赠款项的募集必须通过Safari浏览器访问web页面或是手机短消息完成。

## 22. 法律要件

22.1 应用程序必须遵守所有发布地区当地法律，开发者有义务了解并遵守所有当地法律。

22.2 包含虚假，欺诈或误导性陈述的程序将会被拒绝。

22.3 任何用于招徕、促进或鼓励犯罪或明显鲁莽行为的应用程序将会被拒绝。

22.4 支持非法文件共享的程序将会被拒绝。

22.5 被设计用以非法赌博工具的应用程序（包括点算牌）将会被拒绝。

22.6 具有匿名或恶作剧拨打电话或发送类似短信/彩信功能的程序将会被拒绝。

22.7 任何开发暗中收集用户密码或用户私人数据程序的开发者将会从iOS开发者计划中除名。

22.8 包含非执法机构发布的DUI检查点信息，或者怂恿/协助酒后驾车的应用将会被拒

绝。

22.9 计算药剂用量的应用程序必须由药品制造商或者认可机构发布，比如医院、保险公司以及高校。

22.10. 在未授权的情况下使用iTunes音乐预览的应用程序将会被拒绝。

## 23. Wallet

23.1 Wallet Passes可被用来支付或者接收支付，传递商业信息或者提供验证（比如电影票、飞机票、优惠券以及其他），但把Wallet Passes用于其他用途的应用程序可能会遭到拒绝，并且会被撤销Wallet证书。（2015.9 修正）

（旧版：Passbook Passes可被用来支付或者接收支付，传递商业信息或者提供验证（比如电影票、飞机票、优惠券以及其他），但把Passbook Passes用于其他用途的应用程序可能会遭到拒绝，并且会被撤销Passbook证书。）

23.2 Passes必须包含有效的pass发行人有效的联系资料，否则app将会被拒绝，并且Wallet证书也会被取消。（2015.9 修正）

（旧版：Passes必须包含有效的pass发行人有效的联系资料，否则app将会被拒绝，并且Passbook证书也会被取消。）

23.3 Passes必须经过实体签名，并基于其名字、商标或者品牌进行分发，否则应用程序将会被拒绝，而Wallet证书也可能会被撤销。（2015.9 修正）

（旧版：Passes必须经过实体签名，并基于其名字、商标或者品牌进行分发，否则应用程序将会被拒绝，而Passbook证书也可能会被撤销。）

## 24. 儿童类别

24.1 儿童类别中的应用程序必须包含隐私政策，必须遵守适用的儿童隐私法规。

24.2 儿童类别中的应用程序不允许包括行为广告（比如app内部基于用户行动的服务广告），任何在应用程序中展示的上下文广告必须适合儿童。

24.3 儿童类别中的应用程序必须得到家长许可或使用parental gate才能链接至应用程序外部或进行交易。

24.4 儿童类别中的应用程序必须标明"5岁以下，6-8岁或者9-11岁"。

## 25. 扩展

25.1 包含扩展的应用程序必须遵照 **App Extension Programming Guide** 要求。

25.2 包含扩展的应用程序必须提供某些功能（辅助屏幕，附加设置），否则将会被拒绝。

25.3 如果扩展的视图中包含营销推广、广告或者 IAP 内容，那么包含该扩展的应用将被拒绝。

25.4 键盘扩展必须提供一个切换至下个键盘的方法。

25.5 键盘扩展必须具有离线访问功能，否则将被拒绝。

25.6 键盘扩展必须提供和 **App Extension Programming Guide** 描述一致的数字和十进键盘类型，否则将被拒绝。

25.7 提供键盘扩展的应用必须拥有基本的功能分类和隐私政策，否则将被拒绝。

25.8 提供键盘扩展的应用程序只允许收集用户活动以增强键盘扩展在 iOS 设备上的功能，否则将被拒绝。

## 26. HomeKit

26.1 使用 HomeKit 框架的应用程序必须有提供家庭自动化服务的主要目的。

26.2 使用 HomeKit 框架的应用程序必须在营销文本中说明用途，同时必须提供隐私政策，否则将被拒绝。

26.3 应用程序不允许将从 HomeKit API 收集的数据用于广告宣传或者其他基于使用的数据挖掘。

26.4 出于其他目的使用从 HomeKit API 收集的数据，而不是用于提高用户体验或者家庭自动化功能中硬件/软件性能，这类应用将被拒绝。

## 27. HealthKit 和人体生物学研究

27.1 使用 HealthKit 或者 ResearchKit 框架（出于健康目的用于进行人体生物学研究的框架）的应用程序，必须遵守其所有适用区域的法律，以及 iOS Developer Pro

gram License Agreement 中的 3.3.28 和 3.39 条款。 (2015.3 新增)

27.2 将虚假或者错误的数据写入 HealthKit 的应用程序将会被拒绝。

27.3 使用 HealthKit 框架的应用程序在 iCloud 中储存用户健康信息将会被拒绝。

27.4 应用程序不允许使用通过 HealthKit API 或者进行人体生物学研究收集的用户数据，或者将其泄露给第三方用作广告宣传或者基于使用的数据挖掘目的，除了改善健康、医疗、健康管理以及医学研究。 (2015.3 修正)

27.5 未经用户许可与第三方分享通过 HealthKit API 获得的用户数据的应用程序将会被拒绝。

27.6 使用 HealthKit 框架的应用程序必须在营销文本中说明集成了 Health app，同时必须在 app 用户界面清楚阐释 HealthKit 的功能。

27.7 使用 HealthKit 框架或者进行人体生物学研究的应用程序必须提供隐私政策，否则将会被拒绝。 (2015.3 修正)

27.8 提供诊断、治疗建议，或者控制诊断疾病的硬件，或者治疗疾病的应用程序，若没有根据要求提供书面的监管审批，将会被拒绝。

27.9 收集人体生物学研究相关数据的应用程序必须要获得参与者的许可，对于未成年人，应用程序要得到其父母或者监护人的许可。许可内容必须包括：(a) 研究的性质、目的以及持续性；(b) 参与流程、风险以及受益（福利）；(c) 信息的机密性和数据处理（包括与任何与第三方的共享）；(d) 参与者问题切入点；(e) 取消方法 (2015.3 新增)

27.10 进行健康相关的人体生物学研究的应用程序必须得到独立伦理审查委员会的批准。并根据要求提供证明。 (2015.4 新增)

## 28. TestFlight

28.1 应用程序仅能使用 TestFlight 对以公开发布为目的的应用进行 beta 版测试，且必须遵守完整的 App Review Guidelines。

28.2 当版本中包含的内容或功能有重大变化时，使用 TestFlight 的应用程序必须提交审核。

28.3 使用 TestFlight 的应用程序不允许分发给测试者，以作为任何形式的补偿。

## 29. Apple Pay

29.1 使用Apple Pay的应用程序必须在出售任何商品或者服务之前为用户提供所有材料的购买信息，否则将会被拒绝。使用Apple Pay进行重复付款（译者注：定期付款）的应用程序必须提供最低限度续费期限，付费将持续直至被取消，每个阶段所付款额，费用付款方为客户，以及如何取消等。（2015.3 修正）

29.2 使用Apple Pay的应用程序必须正确使用 Apple Pay Identity Guidelines中的Apple Pay标识和用户界面元素，否则将会被拒绝。（2015.3 修正）

29.3 使用Apple Pay作为购买机制的应用程序所提供的商品或服务不能触犯任何支付地范围内的法律，也不能用作任何非法目的。

29.4 使用Apple Pay的应用程序必须提供隐私政策，否则将会被拒绝。

29.5 只有为了促进或提高商品和服务的交付，或者依照法律要件，使用Apple Pay的应用程序才能与第三方分享通过Apple Pay获得的数据。

### 动态文档

这份文档展现了我们在竭尽所能向您分享我们对提交到App Store的程序的审查方式，我们希望您在开发和提交程序时，这份指南能对您有所帮助。这是一份动态文档，随着新程序和新情况的发生会有所变化。我们会定期更新，以反映这些变化。

感谢您参与到iOS的开发中来。虽然此文档是一份"不该做事宜"的列表，但也请将那份短得多的"必做事宜"列表牢记在心。最重要的是，与我们一道 共同努力让用户感到惊奇和欣喜。用创新方式向他们展示世界，让他们用前所未有的方式与之交流。根据我们的经验，无论是在功能和用户界面上，用户确实会对完善的程序有所响应。更进一步，给他们期望之外的东西，带他们去从未去过的地方。我们愿意提供帮助。

## 效果图

(无)

## 备注

- 文档 - iOS 人机交互指南



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-02-19 | Alfred Jiang | -  |

## 方案名称

时间 - 使用 Benchmarking 精确测量的代码运行时间

## 关键字

Benchmarking \ 测量 \ 代码运行时间

## 需求场景

1. 需要对函数进行运行时间测量时
2. 需要比较两个函数运行时间时

## 参考链接

1. [NSHipster - Benchmarking](#)

## 详细内容

### 1. 使用 **CFAbsoluteTimeGetCurrent**

和 NSDate 或 CFAbsoluteTimeGetCurrent() 偏移量不同的是，mach\_absolute\_time() 和CACurrentMediaTime() 是基于内建时钟的，能够更精确更原子化地测量，并且不会因为外部时间变化而变化（例如时区变化、夏时制、秒突变等）

```

CFTimeInterval startTime = CACurrentMediaTime();
{
    static size_t const count = 1000;
    static size_t const iterations = 10000;

    id object = @"";

    for (size_t i = 0; i < iterations; i++) {
        @autoreleasepool {
            NSMutableArray *mutableArray = [NSMutableArray array];
            for (size_t j = 0; j < count; j++) {
                [mutableArray addObject:object];
            }
        }
    }
}

CFTimeInterval endTime = CACurrentMediaTime();
NSLog(@"Total Runtime: %g s", endTime - startTime);

```

## 2. 使用 `dispatch_benchmark`

`dispatch_benchmark` 是 libdispatch (Grand Central Dispatch) 的一部分。但严肃地说，这个方法并没有被公开声明，所以你必须要自己声明：

```

extern uint64_t dispatch_benchmark(size_t count, void (^block)(void));

```

相比之前的秒计时，毫微秒更加精确，`dispatch_benchmark` 也比手动写循环的 `CFAbsoluteTimeGetCurrent()` 语法结构上看起来更好。

```
uint64_t t = dispatch_benchmark(iterations, ^{
    @autoreleasepool {
        NSMutableArray *mutableArray = [NSMutableArray array];
        for (size_t i = 0; i < count; i++) {
            [mutableArray addObject:object];
        }
    }
});
NSLog(@"%@", [[NSMutableArray array] addObject:] Avg. Runtime: %llu ns", t);
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注                        |
|----|------------|--------------|---------------------------|
| 1  | 2015-03-25 | Alfred Jiang | -                         |
| 2  | 2015-12-23 | Alfred Jiang | -                         |
| 2  | 2016-01-10 | Alfred Jiang | 更新 TTGCDTimerWrapper 实现方式 |

## 方案名称

时间 - 倒计时器的实现

## 关键字

时间 \ 倒计时 \ 计时器 \ 延时执行

## 需求场景

1. 倒计时
2. 延时执行

## 参考链接

1. GitHub - [TTGCDTimerWrapper](#)

## 详细内容

1. 方法定义（注：对 **TTGCDTimerWrapper** 源代码做了少许修改）

TTGCDTimerWrapper.h

```
#import <Foundation/Foundation.h>

@interface TTGCDTimerWrapper : NSObject

//Either a dispatch queue is provided or a default dispatch queue
//will be used. (dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0));
- (id)initWithTargetQueue:(dispatch_queue_t)targetQueue;

//Will perform a block on the provided queue (or default queue)
//after a given interval.
//One can also set it to repeat.
- (void)afterInterval:(NSTimeInterval)interval perform:(void (^)(()))
block repeats:(BOOL) repeats;

//Will cancel the timer and if needed the targetQueue is released.
- (void)cancel;

@end
```

### TTGCDTimerWrapper.m

```
#import "TTGCDTimerWrapper.h"

@implementation TTGCDTimerWrapper {
    dispatch_queue_t _queue;
    dispatch_source_t _timer;
}

- (id)initWithTargetQueue:(dispatch_queue_t)targetQueue {
    self = [super init];

    if(self) {
        _queue = targetQueue ?: dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
        _timer = dispatch_source_create(DISPATCH_SOURCE_TYPE_TIM
```

```

ER, 0, 0, _queue);
}
return self;
}

- (void) afterInterval: (NSTimeInterval)interval perform:(void (^)())block repeats:(BOOL) repeats {
    [self performSync: ^{
        NSTimeInterval timeIntervalInNSec = interval * NSEC_PER_SEC;

        dispatch_source_set_timer(_timer, dispatch_time(DISPATCH_TIME_NOW, timeIntervalInNSec), timeIntervalInNSec, 0);

        dispatch_source_set_event_handler(_timer, ^{
            if(block){
                block();
            }
            if (!repeats) {
                [self cancel];
            }
        });
    }];

    dispatch_resume(_timer);
}];

}

- (void)performSync: (dispatch_block_t)block {
    dispatch_sync(_queue, block);
}

- (void)cancel {
    [self performSync: ^{
        if(_timer) {
            dispatch_source_cancel(_timer);

#ifndef __IPHONE_OS_VERSION_MIN_REQUIRED < 60000
            dispatch_release(_timer);
#endif
        }
    }];
}

```

```
        _timer = NULL;
    }
};

- (void)dealloc {
    [self cancel];
    _queue = NULL;
}

@end
```

## 2. 示例

- 注意点 1：一定要声明为成员变量
- 注意点 2：无论是否 ARC，一定要在 dealloc 中执行 cancel 方法

```
//
//  ViewController.m
//  testproject
//
//  Created by viktyz on 16/1/10.
//  Copyright © 2016年 Alfred Jiang. All rights reserved.
//

#import "ViewController.h"
#import "TTGCDTimerWrapper.h"

@interface ViewController : UIViewController

{
    TTGCDTimerWrapper *timer; //注意点 1：一定要声明为成员变量
}

@end

@implementation ViewController

- (void)dealloc
{
    [timer cancel]; //注意点 2：无论是否 ARC，一定要在 dealloc 中执
```

```
行 cancel 方法
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    timer = [[TTGCDTimerWrapper alloc] initWithTargetQueue:nil];

    [timer afterInterval:5.0
                  perform:^{
                    NSLog(@"After 5 Seconds");
                }
                  repeats:YES];
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

## 效果图

(无)

## 备注

- 注意点 1：一定要声明为成员变量
- 注意点 2：无论是否 ARC，一定要在 dealloc 中执行 cancel 方法
- 时间 - 延时执行解决方案

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-18 | Alfred Jiang | -  |

## 方案名称

时间 - 延时执行解决方案

## 关键字

时间 \ 延时执行

## 需求场景

1. 部分需要延时等待的操作

## 参考链接

1. [博客园 - iOS延时执行的几种方法](#)

## 详细内容

1. 直接使用 `performSelector ````objectiveC`
2. `(void)checkRefreshCatalogViewData { [catalogView performSelector:@selector(refreshDataAfterFiveMinute) withObject:nil afterDelay:kRefreshCatalogViewDelayTime]; }`

//对于代码中调用`performSelector`产生延时操作的代码，如果不是通过`category`方式定义，要在`dealloc`函数里面用`cancelPreviousPerformRequestsWithTarget`取消。因为不取消的话，如果页面在延时的时间内退出，将因为找不到执行函数而崩溃。

- `(void)dealloc { [NSObject cancelPreviousPerformRequestsWithTarget:self`

```
selector:@selector(refreshDataAfterFiveMinute) object:nil]; [super dealloc]; }
```

```
```
```

- Block + performSelector ` ``objectivec @implementation NSObject  
(PerformBlockAfterDelay)
- (void)performBlock:(void (^)(void))block

```
    afterDelay:(NSTimeInterval)delay
```

```
{ block = [block copy]; [self performSelector:@selector(fireBlockAfterDelay):
```

```
    withObject:block
    afterDelay:delay];
```

```
}
```

- (void)fireBlockAfterDelay:(void (^)(void))block { block(); }

```
@end
```

```
3. GCD
```swift
//Swift 解决方案
func delay(delay:Double, closure:() -> ()) {
    dispatch_after(
        dispatch_time(
            DISPATCH_TIME_NOW,
            Int64(delay * Double(NSEC_PER_SEC))
        ),
        dispatch_get_main_queue(), closure)
}

//Objective-C 解决方案
- void RunBlockAfterDelay(NSTimeInterval delay, void (^block)(void))
{
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, NSEC_PER_SEC
*delay),
    dispatch_get_main_queue(), block);
}
```

## 1. 用 animation 的 completion 参数 (不推荐)

```
[UIView animateWithDuration:0.0
    delay:5.0
    options:UIViewAnimationOptionAllowUserInteraction
    animations:^{
        //
    }
    completion:^(BOOL finished) {
        //do stuff here
    }];
}
```

## 2. 使用 NSOperationQueue , 在应用程序的下一个主循环执行 (不推荐)

```
[[NSOperationQueue mainQueue] addOperationWithBlock:aBlock];
```

这个和调用 `performSelector: with afterDelay of 0.0f` 等价

## 效果图

(无)

## 备注

- 时间 - 倒计时器的实现

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

正则表达式 - 在 iOS 开发中使用正则表达式

## 关键字

正则表达式 \ Email 验证 \ ip 验证 \ 域名验证 \ 手机号码验证 \ 输入验证 \ NSString

## 需求场景

1. 需要对字符串进行正则相关的验证时

## 参考链接

1. [RegexKit Framework](#)
2. [RegexKitLite](#)
3. [博客园 - iOS NSPredicate和正则表达式](#)
4. [Regexper : 正则表达式可视化](#)

## 详细内容

### 使用方法

1. 去 [RegexKitLite](#) 下载类库，解压出来会有一个例子包及 2 个文件，其实用到的就这 2 个文件，添加到工程中。
2. 工程中添加 `libcucore.dylib frameworks`。
3. 现在所有的 `NSString` 对象就可以调用 `RegexKitLite` 中的方法了。

```
NSString *email = @“kkk@aaa.com”;
[email isMatchedByRegex:@"\\b([a-zA-Z0-9%_.+\\-]+)@[a-zA-Z0-9\\.\\-]+?\\. [a-zA-Z]{2,6})\\b”];
```

返回 YES，证明是 email 格式，需要注意的是 *RegexKitLite* 用到的正则表达式和 wiki 上的略有区别。

取 string 中 http 端口的例子。

```
NSString *searchString = @"http://www.example.com:8080/index.html";
NSString *regexString = @"\\bhttps://[a-zA-Z0-9\\-\\.]+(?:\\d+)?(?:?:/[a-zA-Z0-9\\-\\.?, '+\\&%$=~*!():@\\\\\\]*))+)?";
NSInteger portInteger = [[searchString stringByMatching:regexString capture:1L] integerValue];
NSLog(@"portInteger: '%ld'", (long)portInteger);

//portInteger: '8080'
```

### 常用正则表达式

匹配双字节字符(包括汉字在内)：[^\u00-\uff]

评注：可以用来计算字符串的长度（一个双字节字符长度计2，ASCII字符计1）

匹配空白行的正则表达式：ns\*r

评注：可以用来删除空白行

匹配HTML标记的正则表达式：`<(S*?)[^>]*>.*?|<.*? />`

评注：网上流传的版本太糟糕，上面这个也仅仅能匹配部分，对于复杂的嵌套标记依旧无能为力

匹配首尾空白字符的正则表达式：`^S*|S*$`

评注：可以用来删除行首行尾的空白字符(包括空格、制表符、换页符等等)，非常有用的表达式

匹配Email地址的正则表达式 : w+([- .]w+)\*@w+([- .]w+)\*.w+([- .]w+)\*

评注：表单验证时很实用

匹配网址URL的正则表达式 : [a-zA-Z]+://[^s]\*

评注：网上流传的版本功能很有限，上面这个基本可以满足需求

匹配帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线) : ^[a-zA-Z][a-zA-Z0-9\_]{4,15}\$

评注：表单验证时很实用

匹配国内电话号码 : d{3}-d{8} | d{4}-d{7}

评注：匹配形式如 0511-4405222 或 021-87888822

匹配腾讯QQ号 : [1-9][0-9]{4,}

评注：腾讯QQ号从10000开始

匹配中国邮政编码 : [1-9]d{5}(?!d)

评注：中国邮政编码为6位数字

匹配身份证 : d{15} | d{18}

评注：中国的身份证为15位或18位

匹配ip地址 : d+.d+.d+.d+ 或 (\d\*\.\.){3}\d\*

评注：提取ip地址时有用

匹配特定数字：

`^-[1-9]d*$ //匹配负整数`

`^-[1-9]d*$ //匹配正整数`

```
^-?[1-9]d*$      //匹配整数  
  
^[1-9]d*|0$     //匹配非负整数（正整数 + 0）  
  
^-?[1-9]d*|0$   //匹配非正整数（负整数 + 0）  
  
^[1-9]d*.d*|0.d*[1-9]d*$    //匹配正浮点数  
  
^-?([1-9]d*.d*|0.d*[1-9]d*)$ //匹配负浮点数  
  
^-?([1-9]d*.d*|0.d*[1-9]d*|0?.0+|0)$ //匹配浮点数  
  
^[1-9]d*.d*|0.d*[1-9]d*|0?.0+|0$      //匹配非负浮点数（正浮点数 + 0）  
  
^-?(-([1-9]d*.d*|0.d*[1-9]d*))|0?.0+|0$ //匹配非正浮点数（负浮点数 + 0）
```

评注：处理大量数据时有用，具体应用时注意修正

匹配特定字符串：

```
^[A-Za-z]+$    //匹配由26个英文字母组成的字符串  
  
^[A-Z]+$      //匹配由26个英文字母的大写组成的字符串  
  
^[a-z]+$      //匹配由26个英文字母的小写组成的字符串  
  
^[A-Za-z0-9]+$ //匹配由数字和26个英文字母组成的字符串  
  
^w+$          //匹配由数字、26个英文字母或者下划线组成的字符串
```

在使用RegularExpressionValidator验证控件时的验证功能及其验证表达式介绍如下：

只能输入数字：“^ [0-9]\*\$”

只能输入n位的数字：“^ d{ n } \$”

只能输入至少n位数字：“^ d{ n, } \$”

只能输入m-n位的数字：“^d{m,n}\$\$”

只能输入零和非零开头的数字：“^(0|[1-9][0-9]\* )\$\$”

只能输入有两位小数的正实数：“^ [0-9]+(.[0-9]{2})?\$\$”

只能输入有1-3位小数的正实数：“^ [0-9]+(.[0-9]{1,3})?\$\$”

只能输入非零的正整数：“^+?[1-9][0-9]\*\$\$”

只能输入非零的负整数：“^- [1-9][0-9]\*\$\$”

只能输入长度为3的字符：“^.{3}\$\$”

只能输入由26个英文字母组成的字符串：“^ [A-Za-z]+\$\$”

只能输入由26个大写英文字母组成的字符串：“^ [A-Z]+\$\$”

只能输入由26个小写英文字母组成的字符串：“^ [a-z]+\$\$”

只能输入由数字和26个英文字母组成的字符串：“^ [A-Za-z0-9]+\$\$”

只能输入由数字、26个英文字母或者下划线组成的字符串：“^w+\$\$”

验证用户密码：“^ [a-zA-Z]w{5,17}\$\$”正确格式为：以字母开头，长度在6-18之间，

只能包含字符、数字和下划线。

验证是否含有^%& ', ;=?\$等字符：“[^%& ', ;=?\$x22]+”

只能输入汉字：“^ [u4e00-u9fa5], {0,}\$\$”

验证Email地址：“^w+[-+.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\*\$\$”

验证InternetURL：“^http://([w-]+.)+[w-]+(/w-./?%&=]\* )?\$\$”

验证电话号码：“^((d{3,4})|d{3,4}-)?d{7,8}\$\$”

正确格式为：“XXXX-XXXXXXX”，“XXXX-XXXXXXXX”，“XXX-XXXXXXX”，

“XXX-XXXXXXXX”，“XXXXXXX”，“XXXXXXX”。

验证身份证号（15位或18位数字）：“^d{15}|d{}18\$”

验证一年的12个月：“^(0?[1-9]|1[0-2])\$”正确格式为：“01”-“09”和“1”“12”

验证一个月的31天：“^((0?[1-9])|((1|2)[0-9])|30|31)\$”

正确格式为：“01”“09”和“1”“31”。

匹配中文字符的正则表达式： [u4e00-u9fa5]

匹配双字节字符(包括汉字在内)：[^x00-xff]

匹配空行的正则表达式：n[s| ]\*r

匹配HTML标记的正则表达式：/<(.\*)>.\*/<(.\*)>/

匹配首尾空格的正则表达式：(^s\*)|(s\*\$)

匹配Email地址的正则表达式：w+([-+.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\*

匹配网址URL的正则表达式：[url=http://%28[w-]+.%29+[w-]+%28/[w]http://([w-]+.)+[w-]+(/[w[/url]-./?%&=]\*)?

### 示例

```
+ (BOOL)isEmailAddress:(NSString *)email
{
    NSString *emailRegex = @"[A-Z0-9a-z._%+-]+@[A-Za-z0-9.-]+\.\[A-Za-z\]{2,4}";
    NSPredicate *emailTest = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", emailRegex];
    return [emailTest evaluateWithObject:email];
}

+ (BOOL)isIPAddress:(NSString *)url
{
    NSString *ipRegex = @"(\d*\.\d*){3}\d*";
    NSPredicate *ipTest = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", ipRegex];
    return [ipTest evaluateWithObject:url];
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-03 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

测试 - iOS 应用测试 Checklist 以及思维导图

## 关键字

测试 \ Checklist \ Test \ 思维导图

## 需求场景

1. 应用测试

## 参考链接

1. [GitHub - app-release-checklist](#)

## 详细内容

### 一、App Development Company Limited

RELEASE AUTHORISATION FORM v1.0

|                    |                       |
|--------------------|-----------------------|
| App name           | <i>insert content</i> |
| Version            | <i>insert content</i> |
| Date of submission | <i>insert content</i> |

This form is to document the testing that has been done on each app version before submitting to the App Store. For each item, indicate *Yes* if the testing has been done, *Not Applicable* if the testing does not apply (eg testing audio for an app that doesn't play any), or *No* if the testing has not been done for another reason.

--

## Internet Connectivity

Test all the data downloading sections of the app by trying them on the appropriate connection type. Consider graceful degradation and failure as well as success conditions.

| Connection                       | N/A | NO | YES |
|----------------------------------|-----|----|-----|
| Wifi                             |     |    |     |
| Edge                             |     |    |     |
| GPRS                             |     |    |     |
| No Network                       |     |    |     |
| Break in Network - use Charles   |     |    |     |
| Server unreachable - timeout     |     |    |     |
| Resumed connect - streaming only |     |    | -   |

--

## Locale

Change device's settings then load the app. Check that dates appear correctly, especially dates from external feeds or services.

| Locale                                           | N/A | NO | YES |
|--------------------------------------------------|-----|----|-----|
| 12 and 24 hour clocks                            |     |    |     |
| Regions: <b>fork and add regions for you</b>     |     |    |     |
| Languages: <b>fork and add languages for you</b> |     |    |     |
| Timezones                                        |     |    |     |
| Daylight Savings Time                            |     |    | -   |

--

## Devices

Run the application through navigations using different devices with different iOS versions and display formats.

| Device                                | N/A | NO | YES |
|---------------------------------------|-----|----|-----|
| iPhone / iPod touch running iOS 5.0   |     |    |     |
| iPhone / iPod touch running iOS 5.1.1 |     |    |     |
| iPhone / iPod touch running iOS 6.0   |     |    |     |
| iPhone / iPod touch running iOS 6.1.3 |     |    |     |
| iPhone / iPod touch running iOS 7.0   |     |    |     |
| Retina iPhone display                 |     |    |     |
| Non-retina iPhone display             |     |    |     |
| iPad 1 running iOS 5.0                |     |    |     |
| iPad 1 running iOS 5.0                |     |    |     |
| iPhone / iPod touch running iOS 5.1.1 |     |    |     |
| iPad running iOS 6.0                  |     |    |     |
| iPad running iOS 6.1.3                |     |    |     |
| iPad running iOS 7.0                  |     |    |     |
| Retina iPad display                   |     |    |     |
| Non-retina iPad display               |     |    |     |
| iPad mini display                     |     |    | -   |

--

## Audio

If app plays audio, perform the following checks. For streaming audio, make sure the checks in the network section above have also been done.

| <b>Audio</b>                                                              | <b>N/A</b> | <b>NO</b> | <b>YES</b> |
|---------------------------------------------------------------------------|------------|-----------|------------|
| Headphones/speaker routing                                                |            |           |            |
| Dock connector audio out routing                                          |            |           |            |
| iPod touch audio routing (consider model without speaker)                 |            |           |            |
| Mute switch functionality (officially it mutes non-user-requested sounds) |            |           |            |
| Audio pause on received phone call                                        |            |           |            |
| Background audio (if supported): playback and multitasking bar controls   |            |           |            |
| Start playing audio when another app is already playing                   |            |           |            |
| Headphone remote for audio control                                        |            |           |            |
| Multitasking screen audio control                                         |            |           | -          |

--

## Video

Streaming video should have been checked in the network tests.

| <b>Video</b>                                    | <b>N/A</b> | <b>NO</b> | <b>YES</b> |
|-------------------------------------------------|------------|-----------|------------|
| User cancels video before playback begins       |            |           |            |
| User cancels video during playback              |            |           |            |
| Video plays to the end                          |            |           |            |
| Video return from full screen                   |            |           |            |
| Dock connector video out                        |            |           |            |
| Video transition between inline and full screen |            |           | -          |

--

## Location

| <b>Location</b>                                                           | <b>N/A</b> | <b>NO</b> | <b>YES</b> |
|---------------------------------------------------------------------------|------------|-----------|------------|
| True GPS                                                                  |            |           |            |
| Wifi location                                                             |            |           |            |
| Cell tower location                                                       |            |           |            |
| Unable to find location                                                   |            |           |            |
| No results returned (e.g. too far from any searchable points of interest) |            |           |            |
| Location services turned off                                              |            |           |            |
| Location services disabled for this app                                   |            |           | -          |

--

## Camera / Video

If app takes pictures or video clips, perform the following checks. For streaming video, make sure the checks in the network section above have also been done.

| <b>Camera / Video</b>                         | <b>N/A</b> | <b>NO</b> | <b>YES</b> |
|-----------------------------------------------|------------|-----------|------------|
| Primary camera photo taken                    |            |           |            |
| Primary camera video captured                 |            |           |            |
| Secondary (user facing) camera taken          |            |           |            |
| Secondary (user facing) video captured        |            |           |            |
| Video recording paused on received phone call |            |           | -          |

--

## Logging

| <b>Logging</b>                              | <b>N/A</b> | <b>NO</b> | <b>YES</b> |
|---------------------------------------------|------------|-----------|------------|
| Logging events to live server               |            |           |            |
| Logging errors (interact with other tests?) |            |           | -          |

--

## User Interface

Test each major view in the app.

| Title                                              | N/A | NO | YES |
|----------------------------------------------------|-----|----|-----|
| Double height status bar (eg in call)              |     |    |     |
| Orientation change                                 |     |    |     |
| Upside-down orientation                            |     |    |     |
| Orientation lock                                   |     |    |     |
| VoiceOver turned on                                |     |    |     |
| Usable by a new user with Screen Curtain turned on |     |    |     |
| Works with Accessibility Zoom turned on            |     |    | -   |

--

## Core Data

| Core Data                                         | N/A | NO | YES |
|---------------------------------------------------|-----|----|-----|
| Validation error in user input                    |     |    |     |
| Validation error in web server input              |     |    |     |
| Test migrations with valid and invalid data files |     |    |     |
| Rollback                                          |     |    | -   |

--

## Installation

| Installation                       | N/A | NO | YES |
|------------------------------------|-----|----|-----|
| Fresh install                      |     |    |     |
| Upgrade from previous live version |     |    |     |
| Upgrade from older live version    |     |    |     |
| Rollback                           |     |    | -   |

--

## Text

| Title                                                | N/A | NO | YES |
|------------------------------------------------------|-----|----|-----|
| Shake to Undo                                        |     |    |     |
| Text selection (including disabled when appropriate) |     |    |     |
| Copy / Paste                                         |     |    |     |
| Editing when keyboard is hidden                      |     |    |     |
| Dictionary / Suggested Word hover                    |     |    | -   |

--

### Third Party Services

All third party services should use production API key and the new app version should be registered in the respective dashboards

| Title                                                                     | N/A | NO | YES |
|---------------------------------------------------------------------------|-----|----|-----|
| Production analytics/tracking API key                                     |     |    |     |
| New app version tracking data available tracking in dashboard             |     |    |     |
| Production crash reporting API key                                        |     |    |     |
| Upload dSYM to crash reporting tool                                       |     |    |     |
| New app version available in crash reporting dashboard                    |     |    |     |
| Push notification service API key                                         |     |    |     |
| New app version added to push service dashboard                           |     |    |     |
| Production App ID for social services (Twitter, Facebook, Instagram, etc) |     |    | -   |

--

### Misc

| Misc                                  | N/A | NO | YES |
|---------------------------------------|-----|----|-----|
| Bluetooth                             |     |    |     |
| Motion                                |     |    |     |
| Tested in Ad Hoc mode                 |     |    |     |
| Version number upgraded               |     |    |     |
| Bundle identifier correct for release |     |    | -   |

--

Sign-off: \_\_\_\_\_

Project role: \_\_\_\_\_

## 二、iOS Testing

### Hardware

#### iPhone

- iPhone 3G
- iPhone 3GS
- iPhone 4 (retina)
- iPhone 4S (retina/dual core)

#### iPod

- 3rd generation (no camera)

#### iPad

- iPad 1st gen
- iPad 2 (dual core)
- The new iPad (retina / dual core)

### UI

#### Accessibility

#### Half pixels

#### Retina display

#### Non-retina display

#### Extended status bar resizing

#### Portrait

#### Landscape

#### Smooth animations

### Functionality

#### Location services

#### Email

Email configured  
    Default to  
    Default subject  
    Default body  
    No email configured

Forced updates

Pull-to-refresh bar  
    Updates timestamp  
    Makes proper network request

Webviews

Network

    3G  
    EDGE  
    Wi-Fi  
    Airplane Mode  
    Simulated poor connection  
        On first launch  
        Error message on timeouts  
        Doesn't block main thread

Data

    Preserved through data model changes

Time Settings

    Properly display all time zones  
    Relative time displays  
    Switching between time zones  
    System time too fast/slow

Internationalization

    Supported languages?  
    Long translated strings fit  
    Localized text in images  
    Region formats

Software

    iOS  
        4.x  
        5.x  
        6.x

    Crash reporting

    Analytics

    Memory Warnings  
        No crashes

View state is preserved

## Upgrades

## Upgrade path from all previous versions

User settings are preserved

## Push notifications

## Credentials

## User preferences

## Backgrounding

App closes in time

App state preserved on next launch

## Gestures

## Audio / Video

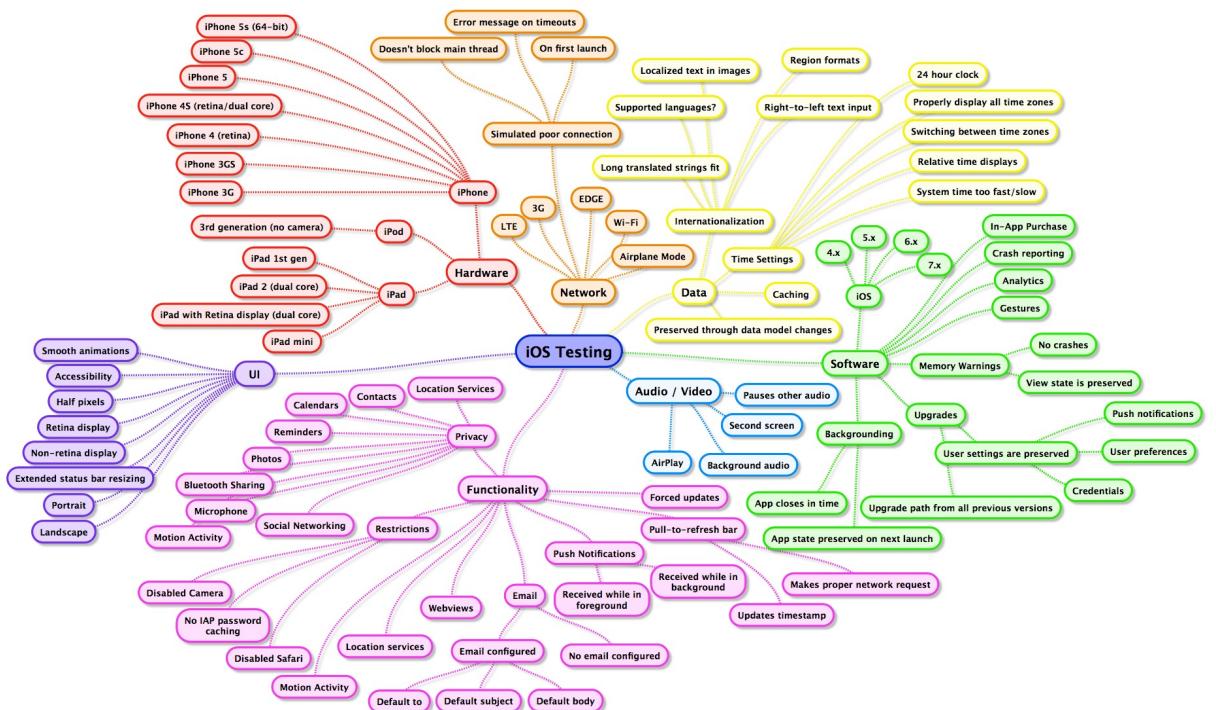
## AirPlay

## Background audio

## Second screen

Pauses other audio

效果图



## 备注

需要根据实际需要和测试人员一起确认适合各 App 的详单



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-27 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

测试 - iOS 自动化测试框架介绍

## 关键字

测试\自动化测试框架

## 需求场景

1. 需要对 iOS 项目进行自动化测试时

## 参考链接

1. iOS 自动化测试工具总览
2. IOS 下自动化测试框架 GHUnit 和 OCMock 初探 [一]
3. APPIUM框架-IOS原生APP自动化测试
4. iOS的OCUnit测试框架
5. 基于instruments-Automation的iOS应用UI自动化测试图文攻略
6. 这些年用过的iOS测试框架

## 详细内容

| 名称     | 简介                                  | 官网        | 参考链接                                              |
|--------|-------------------------------------|-----------|---------------------------------------------------|
| Appium | Appium是一个开源、跨平台的测试框架，可以用来测试原生及混合的移动 | appium.io | 1. IOS、Android 自动化测试框架<br>Appium 概述<br>2. ios 界面自 |

|            |                                                                                                                                                   |                       |                                                 |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-------------------------------------------------|
|            | 端应用。                                                                                                                                              |                       | 自动化工具--Appium                                   |
| Automation | 苹果官方自动化测试工具，包含在 Instruments 中                                                                                                                     | Automating UI Testing | 如何使用 UIAutomation 进行iOS 自动化测试 (Part I) [译] (原文) |
| Calabash   | Calabash is an automated testing technology for Android and iOS native and hybrid applications.                                                   | GitHub - calabash-ios | 【AT】 Calabash-ios 学习                            |
| EarlGrey   | 开源测试框架，用于构建和运行“面向iOS apps源码”的用户界面测试                                                                                                               | GitHub - EarlGrey     | [GitHub - EarlGrey                              |
| Frank      | Frank is 'Selenium for native iOS apps'. It allows you to write automated acceptance tests which verify the functionality of your native iOS app. | GitHub - Frank        | Testing With Frankk                             |
| GHUnit     | 开源测试框架，它可以将测试报告以应用形式可视化输出到设备或模拟器上，也可以以文本形式输出到输出窗口；GHUnit可以测试OCUnit编写的测试用例                                                                         | GitHub - GHUnit       | GHUnit Reference                                |
| KIF        | KIF的全称是“Keep It Functional”，是一款iOS 集成测试框架，来自 square。                                                                                              | GitHub - KIF          | iOS 中使用 KIF 测试 UI [译](原文)                       |
| Kiwi       | Kiwi 是 Objective-C 的单元测试框架，因其简单的接口而著名，基于 Objective-C 编写，只支持 iOS 应用                                                                                | GitHub - Kiwi         | TDD的iOS开发初步以及 Kiwi使用入门                          |
| OCMock     | 开源测试框架，它主要为测试提供Mock对象（伪对象）                                                                                                                        | ocmock.org            | 【OCMock】 使用介绍分析<br>OCMock常见                     |

|          |                                                                        |                                                        | 使用方式                                        |
|----------|------------------------------------------------------------------------|--------------------------------------------------------|---------------------------------------------|
| OCUnit   | 开源测试框架，与 Xcode 工具集成在一起使用非常方便，测试报告以文本形式输出到输出窗口                          | 已更新至 XCTest                                            | 已更新至 XCTest                                 |
| TuneUpJS | 最早的iOS自动化测试工具，以JavaScript扩展库方法提供了很多好用js工具，最重要的是提供了超简洁的单元测试框架和持续继承解决方案。 | <a href="http://www.tuneupjs.org">www.tuneupjs.org</a> |                                             |
| XCTest   | OCUnit 的 Xcode 官方替代测试框架                                                | <a href="#">Testing with Xcode</a>                     | <a href="#">Testing with Xcode 文档 (中文版)</a> |
| ynm3k    | 由半嶧维护的开源测试框架，在 TuneUpJS 基础上加入了UI控件定位的很多方法，使测试脚本更加简单便捷                  | <a href="#">GitHub - ynm3k</a>                         |                                             |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-04-25 | Alfred Jiang | -  |

## 方案名称

测试 - 异步函数的单元测试

## 关键字

测试 \ 单元测试 \ 异步函数 \ delegate \ 代理 \ block

## 需求场景

1. 需要对 Block 异步回调进行结果测试时
2. 需要对 Delegate 异步回调进行结果测试时

## 参考链接

1. [CSDN - iOS中异步函数的单元测试](#)
2. [Stack Overflow - How do I unit test HTTP request and response using NSURLSession in iOS 7.1?](#)

## 详细内容

### 1. Block 回调类异步测试

使用 *XCTTestExpectation*

```

- (void) testDataTask
{
    XCTestExpectation *expectation = [self expectationWithDescription:@"asynchronous request"];

    NSURL *url = [NSURL URLWithString:@"http://www.apple.com"];
    NSURLSessionTask *task = [self.session dataTaskWithURL:url completionHandler:^(NSData *data, NSURLResponse *response, NSError *error) {
        XCTAssertNil(error, @"dataTaskWithURL error %@", error);

        if ([response isKindOfClass:[NSHTTPURLResponse class]])
        {
            NSInteger statusCode = [(NSHTTPURLResponse *) response statusCode];
            XCTAssertEqual(statusCode, 200, @"status code was not 200; was %d", statusCode);
        }

        XCTAssert(data, @"data nil");

        // do additional tests on the contents of the `data` object here, if you want

        // when all done, Fulfill the expectation

        [expectation fulfill];
    }];
    [task resume];

    [self waitForExpectationsWithTimeout:10.0 handler:nil];
}

```

使用 *dispatchsemaphore*

```

- (void) testDataTask
{
    dispatch_semaphore_t semaphore = dispatch_semaphore_create(0);

    NSURL *url = [NSURL URLWithString:@"http://www.apple.com"];
    NSURLSessionTask *task = [self.session dataTaskWithURL:url c
ompletionHandler:^(NSData *data, NSURLResponse *response, NSError
*error) {
        XCTAssertNil(error, @"dataTaskWithURL error %@", error);

        if ([response isKindOfClass:[NSHTTPURLResponse class]])
        {
            NSInteger statusCode = [(NSHTTPURLResponse *) respon
se statusCode];
            XCTAssertEqual(statusCode, 200, @"status code was no
t 200; was %d", statusCode);
        }

        XCTAssert(data, @"data nil");

        // do additional tests on the contents of the `data` obj
ect here, if you want

        // when all done, signal the semaphore

        dispatch_semaphore_signal(semaphore);
    }];
    [task resume];

    long rc = dispatch_semaphore_wait(semaphore, dispatch_time(D
ISPATCH_TIME_NOW, 60.0 * NSEC_PER_SEC));
    XCTAssertEqual(rc, 0, @"network request timed out");
}

```

## 2. Delegate 回调类异步测试

使用 *CFRunLoop*

```
- (void)testExample
{
    NetworkHelper *helper = [[NetworkHelper alloc] initWithDelegate:self];
    [helper getStatusCodeForSite:@"http://www.baidu.com"];
    NSLog(@"----- Waiting -----");
    CFRunLoopRun();
    STAssertTrue(statusCode == 200, @"Can not access this site")
;
    NSLog(@"----- Finished -----");
}

- (void)succeedGotStatusCode:(int)code
{
    statusCode = code;
    CFRunLoopRef runLoopRef = CFRunLoopGetCurrent();
    CFRunLoopStop(runLoopRef);
}

- (void)failedGotStatusCodeWithError:(NSError *)error
{
    // ...
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-10-26 | Alfred Jiang | -  |
| 2  | 2015-12-18 | Alfred Jiang | -  |

## 方案名称

版本兼容 - 7\8 - 无法正确获取 iPad 横竖屏宽高解决方案

## 关键字

版本兼容 \ iPad \ 横竖屏 \ 宽高

## 需求场景

1. 需要获取横竖屏宽高时

## 参考链接

1. [CocoaChina - 主题 : iPad下ios7的app.window一直是竖屏，怎么设置横屏](#)

## 详细内容

```

// 检测版本
inline static int CHECK_IOS() {
    int v = [[getOsVersion() substringToIndex:3] floatValue] * 10
;
    return v;
}

#pragma mark - Use to get corrent screen bounds

- (BOOL)isLandscape {
    UIInterfaceOrientation orientation = [[UIApplication sharedApplication] statusBarOrientation];
    return orientation == UIInterfaceOrientationLandscapeLeft
    || orientation == UIInterfaceOrientationLandscapeRight;
}

- (CGRect)screenBounds {
    BOOL isLandscape = [self isLandscape];
    CGRect screenBounds = [UIScreen mainScreen].bounds;
    float screenWidth = isLandscape ? screenBounds.size.height :
    screenBounds.size.width;
    float screenHeight = isLandscape ? screenBounds.size.width :
    screenBounds.size.height;
    if (CHECK_IOS() >= 80) {
        screenWidth = screenBounds.size.width;
        screenHeight = screenBounds.size.height;
    }
    return CGRectMake(0, 0, screenWidth, screenHeight);
}

```

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-14 | Alfred Jiang | -  |

## 方案名称

特殊控件 - 使用 DBSphereTagCloud 实现标签云效果

## 关键字

特殊控件 \ DBSphereTagCloud

## 需求场景

1. 绘制标签云效果时

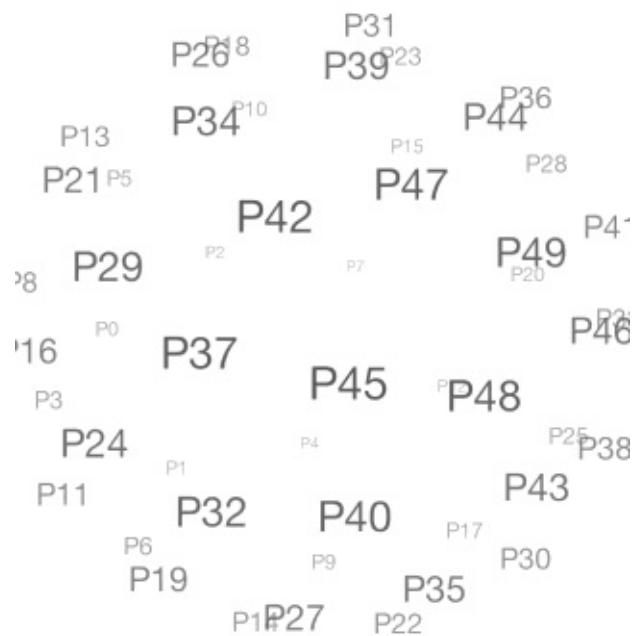
## 参考链接

(见详细内容)

## 详细内容

1. [GitHub - DBSphereTagCloud](#)

## 效果图



备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-16 | Alfred Jiang | -  |

## 方案名称

特殊控件 - 使用 **PSCyclingImageView** 实现自动循环轮播图

## 关键字

特殊控件 \ PSCyclingImageView \ 自动轮播 \ 循环 \ 走马灯

## 需求场景

1. 需要实现轮播图展示时

## 参考链接

1. [GitHub - PSCyclingImageView](#)
2. [GitHub - DCPicScrollView](#)
3. [GitHub - SDCycleScrollView](#)
4. [GitHub - NinaCycleBannerView](#)
5. [GitHub - KDCycleBannerView](#)

## 详细内容

类似 **UITableView** 的使用方法，通过 **PSCyclingImageViewDataSource** 和 **PSCyclingImageViewDelegate** 实现控件的显示和行为获取

### @required

- `(NSInteger)numberOfImagesInCyclingImageView:(nullable PSCyclingImageView *)cyclingImageView`

确定需要显示的图片数目

- (nullable NSString \*)cyclingImageView:(nullable PSCyclingImageView)cyclingImageView imagePathForViewAtIndex:(NSInteger)index

返回需要显示的图片 URL 路径

@optional

- (nullable UIImage \*)cyclingImageView:(nullable PSCyclingImageView)cyclingImageView placeholderImageForViewAtIndex:(NSInteger)index

返回需要显示的图片占位符

- (nullable UIPageControl \*)pageControlInCyclingImageView:(nullable PSCyclingImageView)cyclingImageView

返回需要显示的 UIPageControl, 样式自定义, 默认无 UIPageControl

- (NSTimeInterval)timeIntervalForCyclingImageView:(nullable PSCyclingImageView \*)cyclingImageView

返回自动轮播的时间间隔, 不实现则不会自动轮播

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

特殊控件 - 使用 RecordingCircleOverlayView 实现环形记录仪动画

## 关键字

特殊控件 \ 动画 \ record \ 环形 \ 记录

## 需求场景

1. 录音或者录像显示进度动画

## 参考链接

1. Spark Camera's recording meter
2. GitHub - SparkRecordingCircle

## 详细内容

### 1. RecordingCircleOverlayView.h

```
//  
// RecordingCircleOverlayView.h  
// SparkRecordingCircle  
  
// Created by Sam Page on 1/02/14.  
// Copyright (c) 2014 Sam Page. All rights reserved.  
  
  
#import <UIKit/UIKit.h>  
  
@interface RecordingCircleOverlayView : UIView  
  
- (id)initWithFrame:(CGRect)frame strokeWidth:(CGFloat)strokeWidth insets:(UIEdgeInsets)insets;  
  
@property (nonatomic, assign) CGFloat duration;  
  
@end
```

## 2. RecordingCircleOverlayView.m

```
//  
// RecordingCircleOverlayView.m  
// SparkRecordingCircle  
  
// Created by Sam Page on 1/02/14.  
// Copyright (c) 2014 Sam Page. All rights reserved.  
  
  
#import "RecordingCircleOverlayView.h"  
  
@interface RecordingCircleOverlayView ()  
  
@property (nonatomic, strong) NSMutableArray *progressLayers;  
@property (nonatomic, strong) UIBezierPath *circlePath;  
  
@property (nonatomic, strong) CAShapeLayer *currentProgressLayer  
;  
@property (nonatomic, strong) CAShapeLayer *backgroundLayer;
```

```
@property (nonatomic, assign) CGFloat strokeWidth;
@property (nonatomic, assign, getter = isCircleComplete) BOOL circleComplete;

@end

@implementation RecordingCircleOverlayView

- (id)initWithFrame:(CGRect)frame strokeWidth:(CGFloat)strokeWidth insets:(UIEdgeInsets)insets
{
    if (self = [super initWithFrame:frame])
    {
        self.duration = 45.f;
        self.strokeWidth = strokeWidth;
        self.progressLayers = [NSMutableArray array];

        CGPoint arcCenter = CGPointMake(CGRectGetMidY(self.bounds),
                                         CGRectGetMidX(self.bounds));
        CGFloat radius = CGRectGetMidX(self.bounds) - insets.top
                      - insets.bottom;

        self.circlePath = [UIBezierPath bezierPathWithArcCenter:
                           arcCenter
                           radius:
                           radius
                           startAngle:
                           M_PI
                           endAngle:
                           -M_PI
                           clockwise:
                           NO];
        [self addBackgroundLayer];
    }
    return self;
}

- (void)addBackgroundLayer
{
```

```
    self.backgroundLayer = [CAShapeLayer layer];
    self.backgroundLayer.path = self.circlePath.CGPath;
    self.backgroundLayer.strokeColor = [[UIColor lightGrayColor] CGColor];
    self.backgroundLayer.fillColor = [[UIColor clearColor] CGColor];
    self.backgroundLayer.lineWidth = self.strokeWidth;

    [self.layer addSublayer:self.backgroundLayer];
}

- (void)addNewLayer
{
    CAShapeLayer *progressLayer = [CAShapeLayer layer];
    progressLayer.path = self.circlePath.CGPath;
    progressLayer.strokeColor = [[self randomColor] CGColor];
    progressLayer.fillColor = [[UIColor clearColor] CGColor];
    progressLayer.lineWidth = self.strokeWidth;
    progressLayer.strokeEnd = 0.f;

    [self.layer addSublayer:progressLayer];
    [self.progressLayers addObject:progressLayer];

    self.currentProgressLayer = progressLayer;
}

- (UIColor *)randomColor
{
    CGFloat hue = ( arc4random() % 256 / 256.0 ); // 0.0 to 1.0

    CGFloat saturation = ( arc4random() % 128 / 256.0 ) + 0.5;
    // 0.5 to 1.0, away from white
    CGFloat brightness = ( arc4random() % 128 / 256.0 ) + 0.5;
    // 0.5 to 1.0, away from black
    return [UIColor colorWithHue:hue saturation:saturation brightness:brightness alpha:1.f];
}

- (void)updateAnimations
{
```

```
    CGFloat duration = self.duration * (1.f - [[self.progressLayers firstObject] strokeEnd]);  
    CGFloat strokeEndFinal = 1.f;  
  
    for (CAShapeLayer *progressLayer in self.progressLayers)  
    {  
        CABasicAnimation *strokeEndAnimation = nil;  
        strokeEndAnimation = [CABasicAnimation animationWithKeyPath:@"strokeEnd"];  
        strokeEndAnimation.duration = duration;  
        strokeEndAnimation.fromValue = @(progressLayer.strokeEnd);  
        strokeEndAnimation.toValue = @(strokeEndFinal);  
        strokeEndAnimation.autoreverses = NO;  
        strokeEndAnimation.repeatCount = 0.f;  
  
        CGFloat previousStrokeEnd = progressLayer.strokeEnd;  
        progressLayer.strokeEnd = strokeEndFinal;  
  
        [progressLayer addAnimation:strokeEndAnimation forKey:@"strokeEndAnimation"];  
  
        strokeEndFinal -= (previousStrokeEnd - progressLayer.strokeStart);  
  
        if (progressLayer != self.currentProgressLayer)  
        {  
            CABasicAnimation *strokeStartAnimation = nil;  
            strokeStartAnimation = [CABasicAnimation animationWithKeyPath:@"strokeStart"];  
            strokeStartAnimation.duration = duration;  
            strokeStartAnimation.fromValue = @(progressLayer.strokeStart);  
            strokeStartAnimation.toValue = @(strokeEndFinal);  
            strokeStartAnimation.autoreverses = NO;  
            strokeStartAnimation.repeatCount = 0.f;  
  
            progressLayer.strokeStart = strokeEndFinal;  
  
            [progressLayer addAnimation:strokeStartAnimation forKey:@"strokeStartAnimation"];  
        }  
    }  
}
```

```
Key:@"strokeStartAnimation"];
    }
}

CABasicAnimation *backgroundLayerAnimation = nil;
backgroundLayerAnimation = [CABasicAnimation animationWithKeyPath:@"strokeStart"];
backgroundLayerAnimation.duration = duration;
backgroundLayerAnimation.fromValue = @(self.backgroundLayer.strokeStart);
backgroundLayerAnimation.toValue = @(1.f);
backgroundLayerAnimation.autoreverses = NO;
backgroundLayerAnimation.repeatCount = 0.f;
backgroundLayerAnimation.delegate = self;

self.backgroundLayer.strokeStart = 1.0;

[self.backgroundLayer addAnimation:backgroundLayerAnimation
forKey:@"strokeStartAnimation"];
}

- (void)updateLayerModelsForPresentationState
{
    for (CAShapeLayer *progressLayer in self.progressLayers)
    {
        progressLayer.strokeStart = [progressLayer.presentationLayer strokeStart];
        progressLayer.strokeEnd = [progressLayer.presentationLayer strokeEnd];
        [progressLayer removeAllAnimations];
    }

    self.backgroundLayer.strokeStart = [self.backgroundLayer.presentationLayer strokeStart];
    [self.backgroundLayer removeAllAnimations];
}

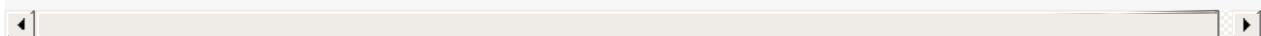
#pragma UIResponder overrides

- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
```

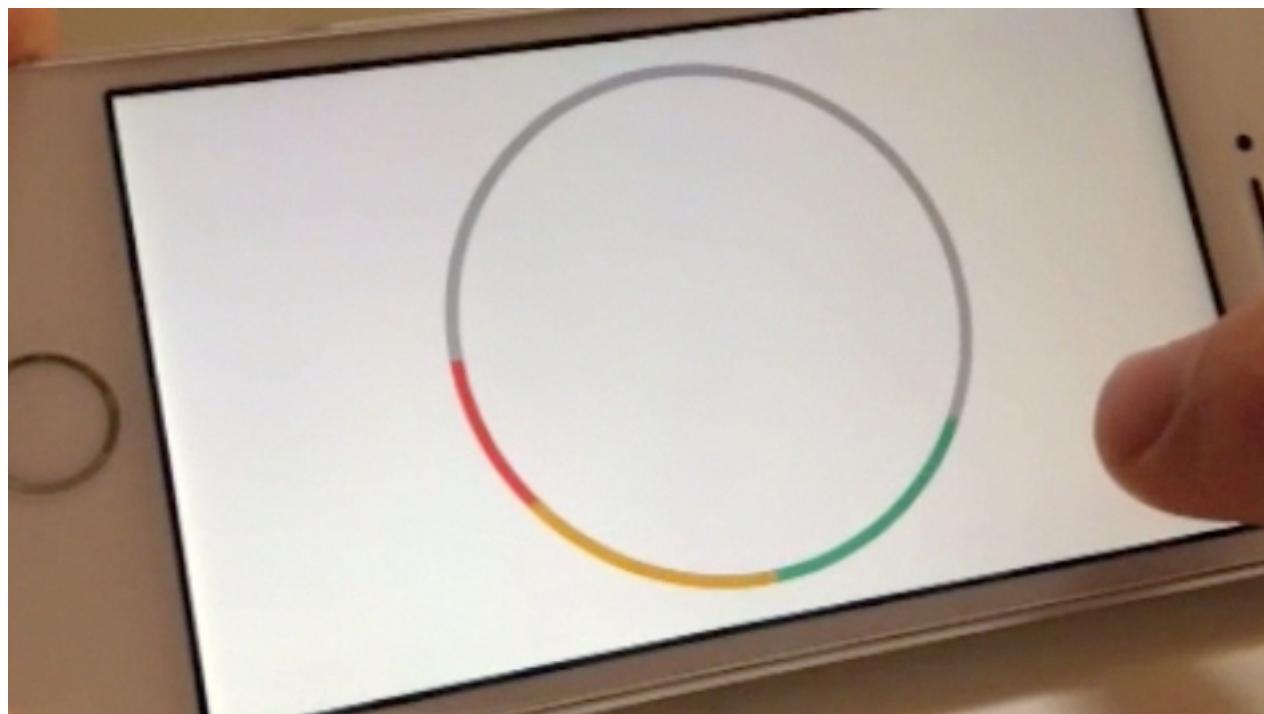
```
{  
    if (self.isCircleComplete == NO)  
    {  
        [self addNewLayer];  
        [self updateAnimations];  
    }  
}  
  
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event  
{  
    if (self.isCircleComplete == NO)  
    {  
        [self updateLayerModelsForPresentationState];  
    }  
}  
  
#pragma mark - CAAnimation Delegate  
  
- (void)animationDidStop:(CAAnimation *)anim finished:(BOOL)flag  
{  
    if (self.isCircleComplete == NO && flag)  
    {  
        self.circleComplete = flag;  
    }  
}  
  
@end
```

### 3. 使用

```
RecordingCircleOverlayView *recordingCircleOverlayView = [[RecordingCircleOverlayView alloc] initWithFrame:CGRectMake(0, 0, 200, 200) strokeWidth:7.f insets:UIEdgeInsetsMake(10.f, 0.f, 10.f, 0.f)];  
recordingCircleOverlayView.autoresizingMask = UIViewAutoresizingFlexibleHeight|UIViewAutoresizingFlexibleWidth;  
recordingCircleOverlayView.duration = 10.f;  
[self.view addSubview:recordingCircleOverlayView];
```



## 效果图



## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

特殊控件 - 使用 SubjectiveCPhotoPanner 实现动作感应显示图片

## 关键字

特殊控件 \ 动画 \ 图片显示 \ 动作

## 需求场景

1. 用于展示一些较大图片，实现较为炫酷的效果

## 参考链接

1. [Facebook Paper's tilting panner](#)
2. [GitHub - SubjectiveCPhotoPanner](#)
3. [GitHub - SubjectiveCPhotoPanner\\_Swift](#)

## 详细内容

### 1. SCImagePanScrollBarView.h

```

//  

//  SCImagePanScrollBarView.h  

//  SubjectiveCImagePan  

//  

//  Created by Sam Page on 16/02/14.  

//  Copyright (c) 2014 Sam Page. All rights reserved.  

//  
  

#import <UIKit/UIKit.h>  
  

@interface SCImagePanScrollBarView : UIView  
  

- (id)initWithFrame:(CGRect)frame edgeInsets:(UIEdgeInsets)edgeI  
nsets;  

- (void)updateWithScrollAmount:(CGFloat)scrollAmount forScrolle  
rWidth:(CGFloat)scrollableWidth inScrollableArea:(CGFloat)scrol  
lableArea;  
  

@end

```

## 2. SCImagePanScrollBarView.m

```

//  

//  SCImagePanScrollBarView.m  

//  SubjectiveCImagePan  

//  

//  Created by Sam Page on 16/02/14.  

//  Copyright (c) 2014 Sam Page. All rights reserved.  

//  
  

#import "SCImagePanScrollBarView.h"  
  

@interface SCImagePanScrollBarView ()  

@property (nonatomic, strong) CAShapeLayer *scrollBarLayer;  

@end  
  

@implementation SCImagePanScrollBarView  
  

- (id)initWithFrame:(CGRect)frame edgeInsets:(UIEdgeInsets)edgeI

```

```

nsets;
{
    if (self = [super initWithFrame:frame])
    {
        UIBezierPath *scrollBarPath = [UIBezierPath bezierPath];
        [scrollBarPath moveToPoint:CGPointMake(edgeInsets.left,
CGRectGetHeight(self.bounds) - edgeInsets.bottom)];
        [scrollBarPath addLineToPoint:CGPointMake(CGRectGetWidth(
self.bounds) - edgeInsets.right, CGRectGetHeight(self.bounds) -
edgeInsets.bottom)];

        CAShapeLayer *scrollBarBackgroundLayer = [CAShapeLayer layer];
        scrollBarBackgroundLayer.path = scrollBarPath.CGPath;
        scrollBarBackgroundLayer.lineWidth = 1.f;
        scrollBarBackgroundLayer.strokeColor = [[[UIColor whiteColor] colorWithAlphaComponent:0.1] CGColor];
        scrollBarBackgroundLayer.fillColor = [[UIColor clearColor] CGColor];

        [self.layer addSublayer:scrollBarBackgroundLayer];

        self.scrollBarLayer = [CAShapeLayer layer];
        self.scrollBarLayer.path = scrollBarPath.CGPath;
        self.scrollBarLayer.lineWidth = 1.f;
        self.scrollBarLayer.strokeColor = [[UIColor whiteColor] CGColor];
        self.scrollBarLayer.fillColor = [[UIColor clearColor] CGColor];
        self.scrollBarLayer.actions = @{@"strokeStart": [NSNull null], @"strokeEnd": [NSNull null]};

        [self.layer addSublayer:self.scrollBarLayer];
    }
    return self;
}

- (void)updateWithScrollAmount:(CGFloat)scrollAmount forScrolle
lWidth:(CGFloat)scrollableWidth inScrollableArea:(CGFloat)scrol
lableArea

```

```

{
    self.scrollBarLayer.strokeStart = scrollAmount * scrollableArea;
    self.scrollBarLayer.strokeEnd = (scrollAmount * scrollableArea) + scrollableWidth;
}

@end

```



### 3. SCImagePanViewController.h

```

//
//  SCImagePanViewController.h
//  SubjectiveCImagePan
//
//  Created by Sam Page on 16/02/14.
//  Copyright (c) 2014 Sam Page. All rights reserved.
//


#import <UIKit/UIKit.h>
#import <CoreMotion/CoreMotion.h>

@interface SCImagePanViewController : UIViewController

- (id)initWithMotionManager:(CMMotionManager *)motionManager;
- (void)configureWithImage:(UIImage *)image;

@end

```

### 4. SCImagePanViewController.m

```

//
//  SCImagePanViewController.m
//  SubjectiveCImagePan
//
//  Created by Sam Page on 16/02/14.
//  Copyright (c) 2014 Sam Page. All rights reserved.
//
```

```
#import "SCImagePanViewController.h"
#import "SCImagePanScrollBarView.h"

@interface SCImagePanViewController () <UIScrollViewDelegate>

@property (nonatomic, strong) CMMotionManager *motionManager;
@property (nonatomic, strong) CADisplayLink *displayLink;

@property (nonatomic, strong) UIScrollView *panningScrollView;
@property (nonatomic, strong) UIImageView *panningImageView;
@property (nonatomic, strong) SCImagePanScrollBarView *scrollBar
View;

@property (nonatomic, assign, getter = isMotionBasedPanEnabled)
BOOL motionBasedPanEnabled;

@end

static CGFloat kMovementSmoothing = 0.3f;
static CGFloat kAnimationDuration = 0.3f;
static CGFloat kRotationMultiplier = 5.f;

@implementation SCImagePanViewController

#pragma mark - init / dealloc

- (id)initWithMotionManager:(CMMotionManager *)motionManager
{
    if (self = [super initWithNibName:nil bundle:nil])
    {
        self.motionManager = motionManager;
        self.motionBasedPanEnabled = YES;
    }
    return self;
}

- (void)dealloc
{
    [_displayLink invalidate];
```

```
[_motionManager stopDeviceMotionUpdates];  
}  
  
#pragma mark - View Lifecycle  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
  
    self.panningScrollView = [[UIScrollView alloc] initWithFrame:  
self.view.bounds];  
    self.panningScrollView.autoresizingMask = UIViewAutoresizingFlexibleWidth|UIViewAutoresizingFlexibleHeight;  
    self.panningScrollView.backgroundColor = [UIColor blackColor];  
    self.panningScrollView.delegate = self;  
    self.panningScrollView.scrollEnabled = NO;  
    self.panningScrollView.alwaysBounceVertical = NO;  
    self.panningScrollView.maximumZoomScale = 2.f;  
    [self.panningScrollView pinchGestureRecognizer addTarget:self  
action:@selector(pinchGestureRecognized:)];  
  
    [self.view addSubview:self.panningScrollView];  
  
    self.panningImageView = [[UIImageView alloc] initWithFrame:s  
elf.view.bounds];  
    self.panningImageView.autoresizingMask = UIViewAutoresizingFlexibleWidth|UIViewAutoresizingFlexibleHeight;  
    self.panningImageView.backgroundColor = [UIColor blackColor];  
    self.panningImageView.contentMode = UIViewContentModeScaleAspectFit;  
  
    [self.panningScrollView addSubview:self.panningImageView];  
  
    self.scrollBarView = [[SCImagePanScrollBarView alloc] initWithFrame:  
self.view.bounds edgeInsets:UIEdgeInsetsMake(0.f, 10.f,  
50.f, 10.f)];  
    self.scrollBarView.autoresizingMask = UIViewAutoresizingFlexibleWidth|UIViewAutoresizingFlexibleHeight;
```

```
self.scrollBarView.userInteractionEnabled = NO;
[self.view addSubview:self.scrollBarView];

self.displayLink = [CADisplayLink displayLinkWithTarget:self
selector:@selector(displayLinkUpdate:)];
[self.displayLink addToRunLoop:[NSRunLoop mainRunLoop] forMode:NSRunLoopCommonModes];

UITapGestureRecognizer *tapGestureRecognizer = [[UITapGestureRecognizer alloc] initWithTarget:self action:@selector(toggleMotionBasedPan:)];
[self.view addGestureRecognizer:tapGestureRecognizer];
}

- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];

    self.panningScrollView.contentOffset = CGPointMake((self.panningScrollView.contentSize.width / 2.f) - (CGRectGetWidth(self.panningScrollView.bounds)) / 2.f,
                                                       (self.panningScrollView.contentSize.height / 2.f) - (CGRectGetHeight(self.panningScrollView.bounds)) / 2.f);

    [self.motionManager startDeviceMotionUpdatesToQueue:[NSOperationQueue mainQueue] withHandler:^(CMDeviceMotion *motion, NSError *error) {
        [self calculateRotationBasedOnDeviceMotionRotationRate:motion];
    }];
}

#pragma mark - Status Bar

- (BOOL)prefersStatusBarHidden
{
    return YES;
}
```

```

#pragma mark - Public

- (void)configureWithImage:(UIImage *)image
{
    self.panningImageView.image = image;
    [self updateScrollViewZoomToMaximumForImage:image];
}

#pragma mark - Motion Handling

- (void)calculateRotationBasedOnDeviceMotionRotationRate:(CMDeviceMotion *)motion
{
    if (self.isMotionBasedPanEnabled)
    {
        CGFloat xRotationRate = motion.rotationRate.x;
        CGFloat yRotationRate = motion.rotationRate.y;
        CGFloat zRotationRate = motion.rotationRate.z;

        if (fabs(yRotationRate) > (fabs(xRotationRate) + fabs(zRotationRate)))
        {
            CGFloat invertedYRotationRate = yRotationRate * -1;

            CGFloat zoomScale = [self maximumZoomScaleForImage:self.panningImageView.image];
            CGFloat interpretedXOffset = self.panningScrollView.contentOffset.x + (invertedYRotationRate * zoomScale * kRotationMultiplier);

            CGPoint contentOffset = [self clampedContentOffsetForHorizontalOffset:interpretedXOffset];

            [UIView animateWithDuration:kMovementSmoothing
                                delay:0.0f
                              options:UIViewAnimationOptionBeginFromCurrentState|UIViewAnimationOptionAllowUserInteraction|UIViewAnimationOptionCurveEaseOut
                               animations:^{
                [self.panningScrollView setCont

```

```
entOffset:contentOffset animated:NO];
} completion:NULL];
}
}

#pragma mark - CADisplayLink

- (void)displayLinkUpdate:(CADisplayLink *)displayLink
{
    CALayer *panningImageViewPresentationLayer = self.panningImage
    view.layer.presentationLayer;
    CALayer *panningScrollViewPresentationLayer = self.panningSc
    rollView.layer.presentationLayer;

    CGFloat horizontalContentOffset = CGRectGetMinX(panningScrol
    lViewPresentationLayer.bounds);

    CGFloat contentWidth = CGRectGetWidth(panningImageVPresen
    tationLayer.frame);
    CGFloat visibleWidth = CGRectGetWidth(self.panningScrollView
    .bounds);

    CGFloat clampedXOffsetAsPercentage = fmax(0.f, fmin(1.f, hor
    izontalContentOffset / (contentWidth - visibleWidth)));

    CGFloat scrollBarWidthPercentage = visibleWidth / contentwid
    th;
    CGFloat scrollableAreaPercentage = 1.0 - scrollBarWidthPerce
    ntage;

    [self.scrollBarView updateWithScrollAmount:clampedXOffsetAsP
    ercentage forScrollableWidth:scrollBarWidthPercentage inScrolle
    rArea:scrollableAreaPercentage];
}

#pragma mark - Zoom toggling

- (void)toggleMotionBasedPan:(id)sender
{
```

```

BOOL motionBasedPanWasEnabled = self.isMotionBasedPanEnabled
;
if (motionBasedPanWasEnabled)
{
    self.motionBasedPanEnabled = NO;
}

[UIView animateWithDuration:kAnimationDuration
    animations:^{
        [self updateViewsForMotionBasedPanEnabled:!motionBasedPanWasEnabled];
    } completion:^(BOOL finished) {
        if (motionBasedPanWasEnabled == NO)
        {
            self.motionBasedPanEnabled = YES;
        }
    }];
}

- (void)updateViewsForMotionBasedPanEnabled:(BOOL)motionBasedPanEnabled
{
    if (motionBasedPanEnabled)
    {
        [self updateScrollViewZoomToMaximumForImage:self.panningImageView.image];
        self.panningScrollView.scrollEnabled = NO;
    }
    else
    {
        self.panningScrollView.zoomScale = 1.f;
        self.panningScrollView.scrollEnabled = YES;
    }
}

#pragma mark - Zooming

- (CGFloat)maximumZoomScaleForImage:(UIImage *)image
{
    return (CGRectGetHeight(self.panningScrollView.bounds) / CGR

```

```
ectGetWidth(self.panningScrollView.bounds)) * (image.size.width
/ image.size.height);
}

- (void)updateScrollViewZoomToMaximumForImage:(UIImage *)image
{
    CGFloat zoomScale = [self maximumZoomScaleForImage:image];

    self.panningScrollView.maximumZoomScale = zoomScale;
    self.panningScrollView.zoomScale = zoomScale;
}

#pragma mark - Helpers

- (CGPoint)clampedContentOffsetForHorizontalOffset:(CGFloat)horizontalOffset;
{
    CGFloat maximumXOffset = self.panningScrollView.contentSize.
width - CGRectGetWidth(self.panningScrollView.bounds);
    CGFloat minimumXOffset = 0.f;

    CGFloat clampedXOffset = fmaxf(minimumXOffset, fmin(horizontalOffset,
maximumXOffset));
    CGFloat centerY = (self.panningScrollView.contentSize.height / 2.f) - (CGRectGetHeight(self.panningScrollView.bounds)) / 2.f;

    return CGPointMake(clampedXOffset, centerY);
}

#pragma mark - Pinch gesture

- (void)pinchGestureRecognized:(id)sender
{
    self.motionBasedPanEnabled = NO;
    self.panningScrollView.scrollEnabled = YES;
}

#pragma mark - UIScrollViewDelegate
```

```
- (UIView *)viewForZoomingInScrollView:(UIScrollView *)scrollView
{
    return self.panningImageView;
}

- (void)scrollViewDidEndZooming:(UIScrollView *)scrollView withView:(UIView *)view atScale:(CGFloat)scale
{
    [scrollView setContentOffset:[self clampedContentOffsetForHorizontalOffset:scrollView.contentOffset.x] animated:YES];
}

- (void)scrollViewDidEndDragging:(UIScrollView *)scrollView willDecelerate:(BOOL)decelerate
{
    if (decelerate == NO)
    {
        [scrollView setContentOffset:[self clampedContentOffsetForHorizontalOffset:scrollView.contentOffset.x] animated:YES];
    }
}

- (void)scrollViewWillBeginDecelerating:(UIScrollView *)scrollView
{
    [scrollView setContentOffset:[self clampedContentOffsetForHorizontalOffset:scrollView.contentOffset.x] animated:YES];
}

@end
```

## 5. 用法

```
CMMotionManager *motionManager = [[CMMotionManager alloc] init];
self.imagePanViewController = [[SCIImagePanViewController alloc]
initWithMotionManager:motionManager];

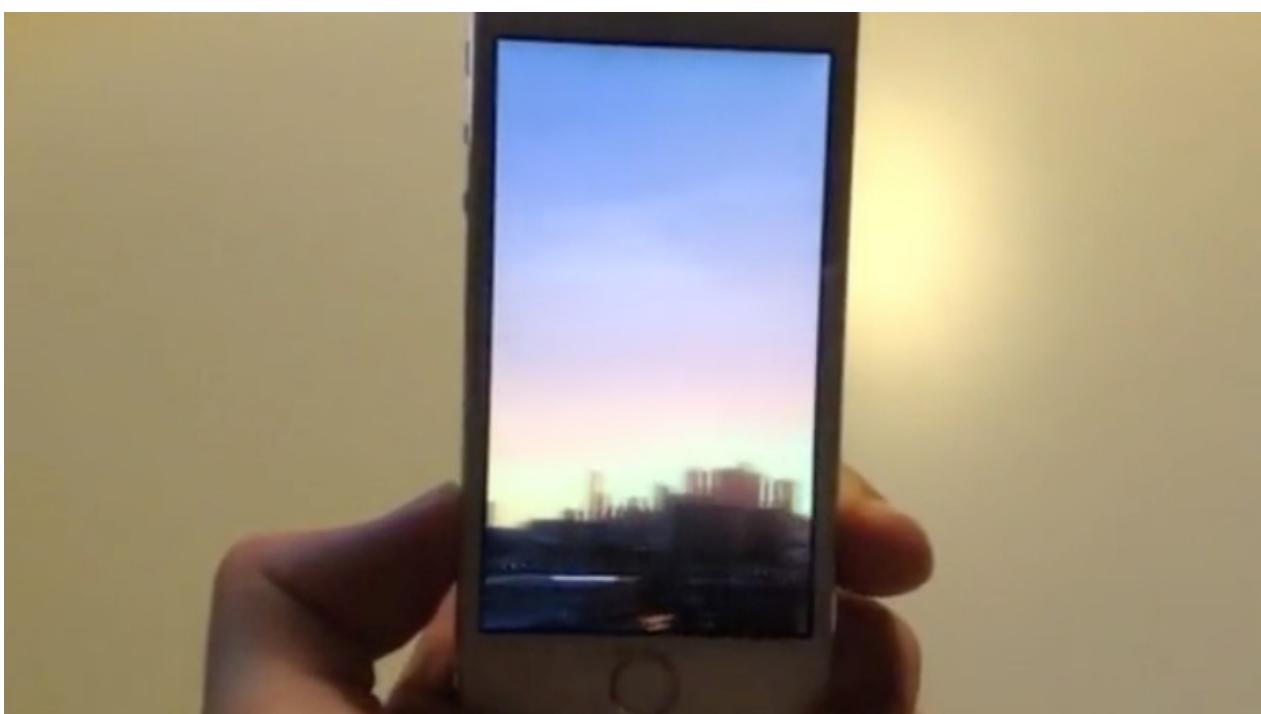
[self.imagePanViewController willMoveToParentViewController:self];
[ self addChildViewController:self.imagePanViewController];
[ self.view addSubview:self.imagePanViewController.view];

self.imagePanViewController.view.frame = self.view.bounds;
self.imagePanViewController.view.autoresizingMask = UIViewAutoresizingFlexibleWidth|UIViewAutoresizingFlexibleHeight;

[self.imagePanViewController didMoveToParentViewController:self];
;

UIImage *panoramaImage = [UIImage imageNamed:@"melbourne.jpg"];
[self.imagePanViewController configureWithImage:panoramaImage];
```

## 效果图



## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-26 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

特殊控件 - 钟表效果实现

## 关键字

特殊控件 \ 时间 \ 表盘 \ 钟面 \ 时钟

## 需求场景

1. 需要时钟效果的页面

## 参考链接

1. [Code4app - Clock View](#)
2. [GitHub - BEMAnalogClock](#)
3. [GitHub - EZClockView](#)

## 详细内容

### 1. Swift 版本 **EZClockView**



静态表盘，需要使用类似倒计时器来不断刷新赋值

```
// Copyright (c) 2015 Benoit Layer
//
// Permission is hereby granted, free of charge, to any person o
btaining a copy
// of this software and associated documentation files (the "Sof
tware"), to deal
// in the Software without restriction, including without limita
tion the rights
// to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell
// copies of the Software, and to permit persons to whom the Sof
tware is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall b
e included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIN
D, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERC
HANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO E
VENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWIS
E, ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN
// THE SOFTWARE.

import UIKit

@IBDesignable
public class EZClockView: UIView {

    // MARK: - Properties
    private var faceView: UIView = UIView()
```

```
private var centerView: UIView = UIView()
private var handHours: UIView = UIView()
private var handMinutes: UIView = UIView()
private var handSeconds: UIView = UIView()

private var hourProperty: Int = 0
private var minuteProperty: Int = 0
private var secondProperty: Int = 0

// MARK: animation
/// Set the animation duration (the view is animated when calling the setTime methods)
public var animationDuration: NSTimeInterval = 0.3

// MARK: Time
/// Set this property to change the hour hand position.
@IBInspectable public var hours: Int {
    get {
        return hourProperty
    }
    set {
        hourProperty = newValue
        updateHands()
    }
}
/// Set this property to change the minutes hand position.
@IBInspectable public var minutes: Int {
    get {
        return minuteProperty
    }
    set {
        minuteProperty = newValue
        updateHands()
    }
}
/// Set this property to change the seconds hand position.
@IBInspectable public var seconds: Int {
    get {
        return secondProperty
    }
}
```

```

    set {
        secondProperty = newValue
        updateHands()
    }

}

// MARK: Face
/// Defines the background color of the face. Defaults to white.
@IBInspectable public var faceBackgroundColor: UIColor = UIColor.whiteColor() { didSet { faceView.backgroundColor = faceBackgroundColor } }
/// Defines the border color of the face. Defaults to black.
@IBInspectable public var faceBorderColor: UIColor = UIColor.blackColor()
/// Defines the border width of the face. Defaults to 2.
@IBInspectable public var faceBorderWidth: CGFloat = 2.0

// MARK: Center disc
/// Defines the color of the rounded part in the middle of the face, over the needles. Default is red.
@IBInspectable public var centerColor: UIColor = UIColor.redColor() { didSet { centerView.backgroundColor = centerColor } }
/// Desfines the width of the center circle. Default is 3.0.
@IBInspectable public var centerRadius: CGFloat = 3.0 { didSet { setupCenterView() } }
/// Defines the width for the border of this center part. Default is 1.0.
@IBInspectable public var centerBorderWidth: CGFloat = 1.0 { didSet { centerView.layer.borderWidth = centerBorderWidth } }
/// Defines the color for the border of this center part. Default is red.
@IBInspectable public var centerBorderColor: UIColor = UIColor.redColor() { didSet { centerView.layer.borderColor = centerBorderColor.CGColor } }

// MARK: Hour hand
/// Defines the color for the hours hand. Default to black.
@IBInspectable public var hoursColor: UIColor = UIColor.blackColor() { didSet { handHours.backgroundColor = hoursColor } }

```

```

    /// Defines the length of the hours hand. It is represented
    as a ratio of the radius of the face. Default to 0.5.
    @IBInspectable public var hoursLength: CGFloat = 0.5 {
        didSet { setupHand(handHours, lengthRatio: hoursLength,
        thickness: hoursThickness, offset: hoursOffset) }
    }
    /// Defines the thickness of the hours hand. Default is 4.
    @IBInspectable public var hoursThickness: CGFloat = 4 {
        didSet { setupHand(handHours, lengthRatio: hoursLength,
        thickness: hoursThickness, offset: hoursOffset) }
    }
    /// Defines the distance by which the hours hand will overla-
    p over the center of the face. Default is 2.
    @IBInspectable public var hoursOffset: CGFloat = 2 {
        didSet { setupHand(handHours, lengthRatio: hoursLength,
        thickness: hoursThickness, offset: hoursOffset) }
    }

    // MARK: Minute hand
    /// Defines the color for the minutes hand. Default to black.

    @IBInspectable public var minutesColor: UIColor = UIColor.bl-
    ackColor() { didSet { handMinutes.backgroundColor = minutesColor
    } }
    /// Defines the length of the minutes hand. It is represente-
    d as a ratio of the radius of the face. Default to 0.7.
    @IBInspectable public var minutesLength: CGFloat = 0.7 {
        didSet { setupHand(handMinutes, lengthRatio: minutesLeng-
        th, thickness: minutesThickness, offset: minutesOffset) }
    }
    /// Defines the thickness of the minutes hand. Default is 2.
    @IBInspectable public var minutesThickness: CGFloat = 2 {
        didSet { setupHand(handMinutes, lengthRatio: minutesLeng-
        th, thickness: minutesThickness, offset: minutesOffset) }
    }
    /// Defines the distance by which the minutes hand will over-
    lap over the center of the face. Default is 2.
    @IBInspectable public var minutesOffset: CGFloat = 2 {
        didSet { setupHand(handMinutes, lengthRatio: minutesLeng-
        th, thickness: minutesThickness, offset: minutesOffset) }
    }

```

```

}

// MARK: Second hand
/// Defines the color for the seconds hand. Default to red.
@IBInspectable public var secondsColor: UIColor = UIColor.redColor() {
    didSet { handSeconds.backgroundColor = secondsColor }
}

/// Defines the length of the seconds hand. It is represented as a ratio of the radius of the face. Default to 0.8.
@IBInspectable public var secondsLength: CGFloat = 0.8 {
    didSet { setupHand(handSeconds, lengthRatio: secondsLength, thickness: secondsThickness, offset: secondsOffset) }
}

/// Defines the thickness of the seconds hand. Default is 1.
@IBInspectable public var secondsThickness: CGFloat = 1 {
    didSet { setupHand(handSeconds, lengthRatio: secondsLength, thickness: secondsThickness, offset: secondsOffset) }
}

/// Defines the distance by which the seconds hand will overlap over the center of the face. Default is 2.
@IBInspectable public var secondsOffset: CGFloat = 2 {
    didSet { setupHand(handSeconds, lengthRatio: secondsLength, thickness: secondsThickness, offset: secondsOffset) }
}

// MARK: - Public methods
/**
 Set the time the clock will display. You can animate it or not.

 :param: h The hour to set
 :param: m The minute to set
 :param: s The second to set
 :param: animated Whether or not the change should be animated (default to false).
 */
public func setTime(#h: Int, #m: Int, #s: Int, animated: Bool = false) {
    hourProperty = h
    minuteProperty = m
}

```

```

        secondProperty = s
        updateHands(animated: animated)
    }

    /**
     Set the time the clock will display directly by using an NSD
     ate instance.

     :param: date The date to display. Only hours, minutes, and s
     econds, will be taken into account.
     :param: animated Whether or not the change should be animate
     d (default to false).
    */
    public func setTime(date: NSDate, animated: Bool = false) {
        let components = NSCalendar.currentCalendar().components
        ((.HourCalendarUnit | .MinuteCalendarUnit | .SecondCalendarUnit)
        , fromDate: date)
        hourProperty = components.hour
        minuteProperty = components.minute
        secondProperty = components.second
        updateHands(animated: animated)
    }

    public override func layoutSubviews() {
        super.layoutSubviews()

        let clockRadius = min(self.bounds.size.width, self.bound
        s.size.height)

        // Reset all transforms
        setupHand(handHours, lengthRatio: hoursLength, thicknes
        s: hoursThickness, offset: hoursOffset)
        setupHand(handMinutes, lengthRatio: minutesLength, thicknes
        s: minutesThickness, offset: minutesOffset)
        setupHand(handSeconds, lengthRatio: secondsLength, thicknes
        s: secondsThickness, offset: secondsOffset)

        setupCenterView()

        handHours.backgroundColor = hoursColor
    }
}

```

```

        handMinutes.backgroundColor = minutesColor
        handSeconds.backgroundColor = secondsColor

        faceView.frame.size = CGSize(width: clockRadius, height:
clockRadius)
        faceView.center = CGPointMake(CGRectGetMidX(self.bounds)
, CGRectGetMidY(self.bounds))
        faceView.layer.cornerRadius = clockRadius/2.0
        faceView.backgroundColor = faceBackgroundColor
        faceView.layer.borderWidth = faceBorderWidth
        faceView.layer.borderColor = faceBorderColor.CGColor

    setTime(h: hours, m: minutes, s: seconds)
}

public override func willMoveToSuperview(newSuperview: UIView?
) {
    if (faceView.superview == nil) {

        self.backgroundColor = UIColor.clearColor()

        self.addSubview(faceView)
        self.addSubview(handHours)
        self.addSubview(handMinutes)
        self.addSubview(handSeconds)
        self.addSubview(centerView)
    }
}

// MARK: - Private methods
private func setupHand(hand: UIView, lengthRatio: CGFloat, t
hickness: CGFloat, offset: CGFloat) {
    hand.transform = CGAffineTransformIdentity
    hand.layer.allowsEdgeAntialiasing = true

    let clockRadius = min(self.bounds.size.width, self.bound
s.size.height)
    let handLength = (clockRadius/2.0) * CGFloat(lengthRatio
)
}

```

```

        let anchorX: CGFloat = 0.5
        let anchorY: CGFloat = 1.0 - (offset/handLength)
        hand.layer.anchorPoint = CGPoint(x: anchorX, y: anchorY)

        let centerInParent = CGPointMake(CGRectGetMidX(self.bounds),
                                         CGRectGetMidY(self.bounds))
        hand.frame = CGRectMake(centerInParent.x-(thickness/2),
                               centerInParent.y - handLength + offset, thickness, handLength)

        // Replace the hand at appropriate position
        updateHands()
    }

private func setupCenterView() {
    centerView.bounds = CGRect(origin: CGPointMakeZero, size: CGSize(width: centerRadius*2, height: centerRadius*2))
    centerView.center = CGPointMake(CGRectGetMidX(self.bounds),
                                   CGRectGetMidY(self.bounds))
    centerView.layer.cornerRadius = centerRadius
    centerView.backgroundColor = centerColor
    centerView.layer.borderColor = centerBorderColor.CGColor
    centerView.layer.borderWidth = centerBorderWidth
}

private func updateHands(animated: Bool = false) {
    // Put everything in seconds to have ratios
    let hoursInSeconds = (hours%12)*3600
    let minutesInSeconds = (minutes%60)*60
    let secondsInSeconds = (seconds%60)

    let hoursRatio = CGFloat(hoursInSeconds + minutesInSeconds + secondsInSeconds) / 43200.0
    let minutesRatio = CGFloat(minutesInSeconds + secondsInSeconds) / 3600.0
    let secondsRatio = CGFloat(secondsInSeconds) / 60.0

    if (animated) {
        UIView.animateWithDuration(animationDuration) {
            self.handSeconds.transform = CGAffineTransformMa
    }
}

```

```

keRotation(CGFloat(2*M_PI)*secondsRatio)
    self.handMinutes.transform = CGAffineTransformMakeRotation(CGFloat(2*M_PI)*minutesRatio)
        self.handHours.transform = CGAffineTransformMakeRotation(CGFloat(2*M_PI)*hoursRatio)
    }
} else {
    handSeconds.transform = CGAffineTransformMakeRotation(CGFloat(2*M_PI)*secondsRatio)
    handMinutes.transform = CGAffineTransformMakeRotation(CGFloat(2*M_PI)*minutesRatio)
    handHours.transform = CGAffineTransformMakeRotation(CGFloat(2*M_PI)*hoursRatio)
}
}

```

## 2. Objective-C 版本 Clock View

动态表盘，较早版本，MRC编写，需要设置 -fno-objc-arc

1. ClockView.h ````objectivec // // // clock // // Created by Ignacio Enriquez Gutierrez on 1/31/11. // Copyright 2011 Nacho4D. All rights reserved. // See the file License.txt for copying permission. //

**import**

**import**

@interface ClockView : UIView {

```

CALayer *containerLayer;
CALayer *hourHand;
CALayer *minHand;
CALayer *secHand;
NSTimer *timer;

```

```

}

//basic methods

• (void)start;
• (void)stop;

//customize appearance

• (void)setHourHandImage:(CGImageRef)image;
• (void)setMinHandImage:(CGImageRef)image;
• (void)setSecHandImage:(CGImageRef)image;
• (void)setClockBackgroundImage:(CGImageRef)image;

//to customize hands size: adjust following values in .m file
//HOURS_HAND_LENGTH //MIN_HAND_LENGTH //SEC_HAND_LENGTH
//HOURS_HAND_WIDTH //MIN_HAND_WIDTH //SEC_HAND_WIDTH

@end

```

```

2. ClockView.m
```objective-c
//
//  ClockView.m
//  clock
//
//  Created by Ignacio Enriquez Gutierrez on 1/31/11.
//  Copyright 2011 Nacho4D. All rights reserved.
//  See the file License.txt for copying permission.
//


#import "ClockView.h"

@implementation ClockView

#pragma mark - Public Methods

- (void)start
{
    timer = [NSTimer scheduledTimerWithTimeInterval:1.0 target:s

```

```
elf selector:@selector(updateClock:) userInfo:nil repeats:YES];
}

- (void)stop
{
    [timer invalidate];
    timer = nil;
}

//customize appearance
- (void)setHourHandImage:(CGImageRef)image
{
    if (image == NULL) {
        hourHand.backgroundColor = [UIColor grayColor].CGColor;
        hourHand.cornerRadius = 3;
    }else{
        hourHand.backgroundColor = [UIColor clearColor].CGColor;
        hourHand.cornerRadius = 0.0;
    }
    hourHand.contents = (id)image;
}

- (void)setMinHandImage:(CGImageRef)image
{
    if (image == NULL) {
        minHand.backgroundColor = [UIColor grayColor].CGColor;
    }else{
        minHand.backgroundColor = [UIColor clearColor].CGColor;
    }
    minHand.contents = (id)image;
}

- (void)setSecHandImage:(CGImageRef)image
{
    if (image == NULL) {
        secHand.backgroundColor = [UIColor whiteColor].CGColor;
        secHand.borderWidth = 1.0;
        secHand.borderColor = [UIColor redColor].CGColor;
    }else{
```

```

        secHand.backgroundColor = [UIColor clearColor].CGColor;
        secHand.borderWidth = 0.0;
        secHand.borderColor = [UIColor clearColor].CGColor;
    }
    secHand.contents = (id)image;
}

- (void)setClockBackgroundImage:(CGImageRef)image
{
    if (image == NULL) {
        containerLayer.borderColor = [UIColor blackColor].CGColor;
        containerLayer.borderWidth = 1.0;
        containerLayer.cornerRadius = 5.0;
    }else{
        containerLayer.borderColor = [UIColor clearColor].CGColor;
        containerLayer.borderWidth = 0.0;
        containerLayer.cornerRadius = 0.0;
    }
    containerLayer.contents = (id)image;
}

#pragma mark - Private Methods

//Default sizes of hands:
//in percentage (0.0 - 1.0)
#define HOURS_HAND_LENGTH 0.65
#define MIN_HAND_LENGTH 0.75
#define SEC_HAND_LENGTH 0.8
//in pixels
#define HOURS_HAND_WIDTH 1
#define MIN_HAND_WIDTH 1
#define SEC_HAND_WIDTH 1

float Degrees2Radians(float degrees) { return degrees * M_PI / 180; }

//timer callback
- (void) updateClock:(NSTimer *)theTimer{

```

```

NSDateComponents *dateComponents = [[NSCalendar currentCalendar] components:(NSCalendarUnitHour | NSCalendarUnitMinute | NSCalendarUnitSecond) fromDate:[NSDate date]];
NSInteger seconds = [dateComponents second];
NSInteger minutes = [dateComponents minute];
NSInteger hours = [dateComponents hour];
//NSLog(@"raw: hours:%d min:%d secs:%d", hours, minutes, seconds);
if (hours > 12) hours -=12; //PM

//set angles for each of the hands
CGFloat secAngle = Degrees2Radians(seconds/60.0*360);
CGFloat minAngle = Degrees2Radians(minutes/60.0*360);
CGFloat hourAngle = Degrees2Radians(hours/12.0*360) + minAngle/12.0;

//reflect the rotations + 180 degrees since CALayers coordinate system is inverted
secHand.transform = CATransform3DMakeRotation (secAngle+M_PI, 0, 0, 1);
minHand.transform = CATransform3DMakeRotation (minAngle+M_PI, 0, 0, 1);
hourHand.transform = CATransform3DMakeRotation (hourAngle+M_PI, 0, 0, 1);
}

#pragma mark - Overrides

- (void) layoutSubviews
{
    [super layoutSubviews];

    containerLayer.frame = CGRectMake(0, 0, self.frame.size.width, self.frame.size.height);

    float length = MIN(self.frame.size.width, self.frame.size.height)/2;
    CGPoint c = CGPointMake(self.frame.size.width/2, self.frame.size.height/2);
}

```

```

hourHand.position = minHand.position = secHand.position = c;

CGFloat w, h;

if (hourHand.contents == NULL){
    w = HOURS_HAND_WIDTH;
    h = length*HOURS_HAND_LENGTH;
}else{
    w = CGImageGetWidth((CGImageRef)hourHand.contents);
    h = CGImageGetHeight((CGImageRef)hourHand.contents);
}
hourHand.bounds = CGRectMake(0,0,w,h);

if (minHand.contents == NULL){
    w = MIN_HAND_WIDTH;
    h = length*MIN_HAND_LENGTH;
}else{
    w = CGImageGetWidth((CGImageRef)minHand.contents);
    h = CGImageGetHeight((CGImageRef)minHand.contents);
}
minHand.bounds = CGRectMake(0,0,w,h);

if (secHand.contents == NULL){
    w = SEC_HAND_WIDTH;
    h = length*SEC_HAND_LENGTH;
}else{
    w = CGImageGetWidth((CGImageRef)secHand.contents);
    h = CGImageGetHeight((CGImageRef)secHand.contents);
}
secHand.bounds = CGRectMake(0,0,w,h);

hourHand.anchorPoint = CGPointMake(0.5,0.0);
minHand.anchorPoint = CGPointMake(0.5,0.0);
secHand.anchorPoint = CGPointMake(0.5,0.0);
containerLayer.anchorPoint = CGPointMake(0.5, 0.5);
}

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
}

```

```
if (self) {  
  
    containerLayer = [[[CALayer layer] retain];  
    hourHand = [[[CALayer layer] retain];  
    minHand = [[[CALayer layer] retain];  
    secHand = [[[CALayer layer] retain];  
  
    //default appearance  
    [self setClockBackgroundImage:NULL];  
    [self setHourHandImage:NULL];  
    [self setMinHandImage:NULL];  
    [self setSecHandImage:NULL];  
  
    //add all created sublayers  
    [containerLayer addSublayer:hourHand];  
    [containerLayer addSublayer:minHand];  
    [containerLayer addSublayer:secHand];  
    [self.layer addSublayer:containerLayer];  
}  
return self;  
}  
  
- (void)dealloc  
{  
    [self stop];  
    [hourHand release];  
    [minHand release];  
    [secHand release];  
    [containerLayer release];  
  
    [super dealloc];  
}  
  
@end
```

### 3. Objective-C 版本 [BEMAnalogClock](#)



功能较为完善版本，可以手动拨动指针实现动态修改时间，如果需要实现较为复杂的时钟可考虑使用该版本

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-17 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

特殊控件 - 雷达图（蛛网图、五行图）解决方案

## 关键字

特殊控件 \ 雷达图 \ 蛛网图 \ 五行图

## 需求场景

1. 绘制雷达图需求时

## 参考链接

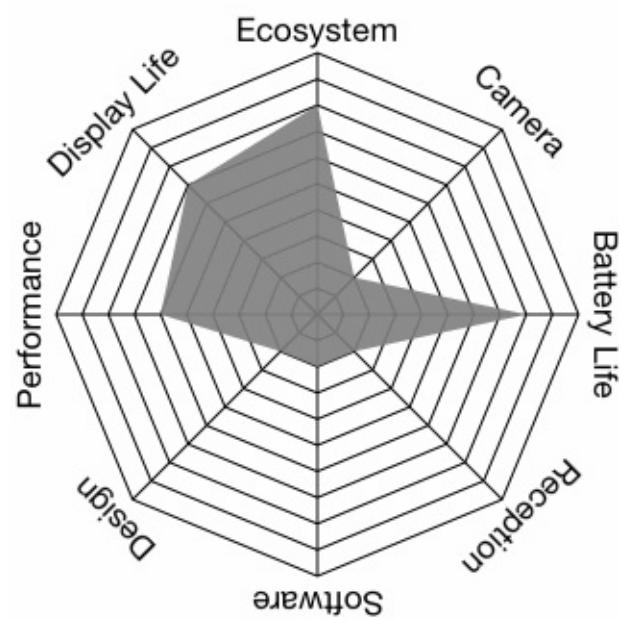
(见详细内容)

## 详细内容

1. [GitHub - BTSpiderPlotterView](#)

2. [GitHub - JYRadarChart](#)

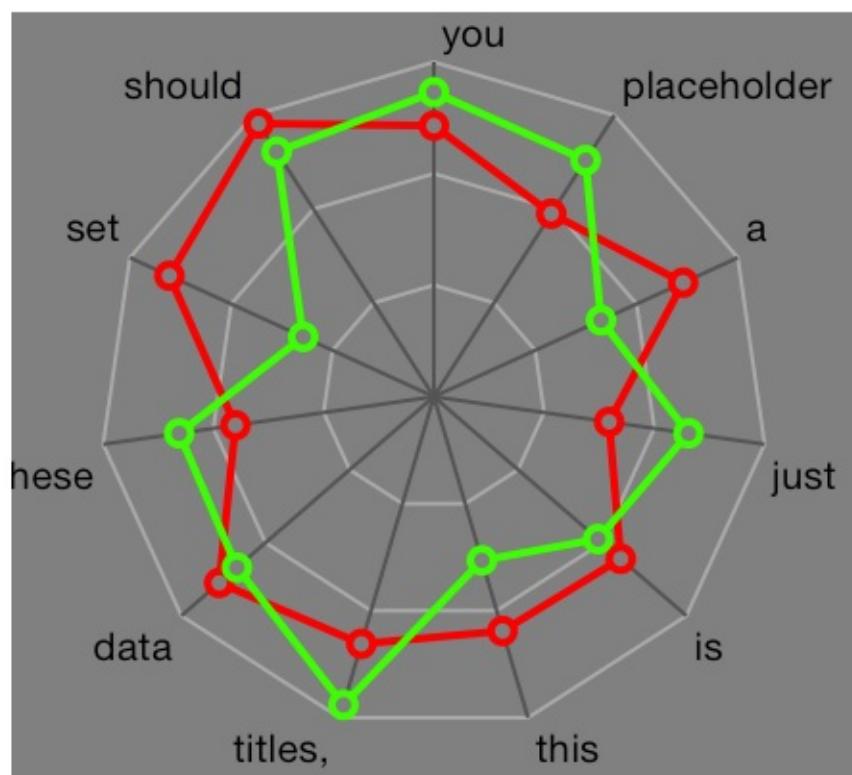
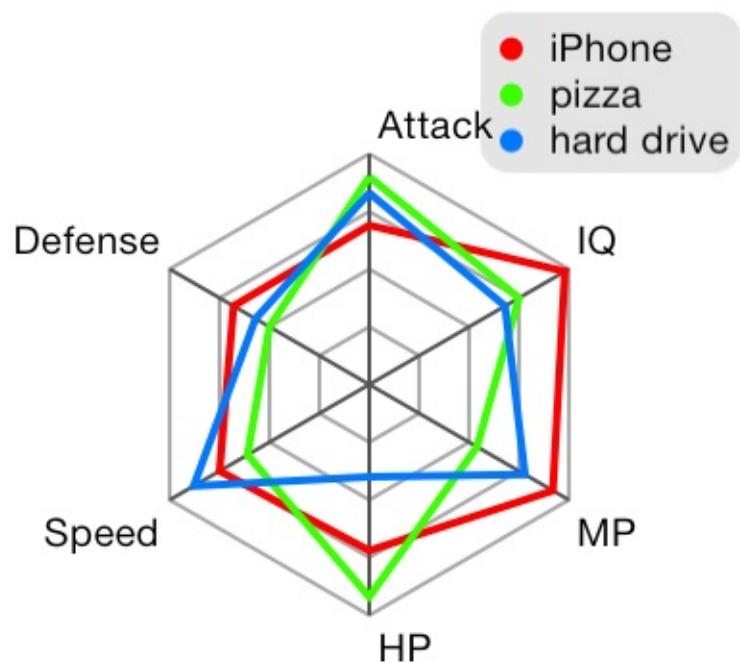
## 效果图



Tap anywhere

Carrier WiFi

12:23 AM



## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-20 | Alfred Jiang | -  |

## 方案名称

界面设计 - iOS 设计相关图标与图像尺寸

## 关键字

界面设计 \ 文档 \ 图标尺寸 \ 图像尺寸 \ 系统默认尺寸

## 需求场景

1. 需要确定 iOS 设计尺寸时

## 参考链接

1. [Deeper - 图标和图像尺寸](#)
2. [Apple documentation - Icon and Image Sizes](#)

## 详细内容

### 参数

| Asset              | iPhone 6 Plus(@3x) | iPhone 6 and iPhone 5(@2x) | iPhone 4s(@2x) | iPad2 and iPad mini2(@2x) | iPad and iPad mini(@1x) |
|--------------------|--------------------|----------------------------|----------------|---------------------------|-------------------------|
| 应用图标<br>(需要适用所有程序) | 180 x 180          | 120 x 120                  | 120 x 120      | 152 x 152                 | 76 x 76                 |
| App Store          |                    |                            |                |                           |                         |

| 应用图标<br>(需要适用所有程序)              | 1024 x 1024                                  | 1024 x 1024                                                                        | 1024 x 1024              | 1024 x 1024                                  | 1024 x 1024                                |
|---------------------------------|----------------------------------------------|------------------------------------------------------------------------------------|--------------------------|----------------------------------------------|--------------------------------------------|
| 启动页或图像(需要适用所有程序)                | 竖屏：<br>1242 x 2208<br><br>横屏：<br>2208 x 1242 | iPhone 6：<br>竖屏：<br>750 x 1334<br>横屏：<br>1334 x 750<br><br>iPhone 5：<br>640 x 1136 | 640 x 960                | 竖屏：<br>1536 x 2048<br><br>横屏：<br>2048 x 1536 | 竖屏：<br>768 x 1024<br><br>横屏：<br>1024 x 768 |
| Spotlight 搜索结果图标(推荐)            | 120 x 120                                    | 80 x 80                                                                            | 80 x 80                  | 80 x 80                                      | 40 x 40                                    |
| 设置图标(推荐)                        | 87 x 87                                      | 58 x 58                                                                            | 58 x 58                  | 58 x 58                                      | 29 x 29                                    |
| 工具栏和导航栏图标(可选)                   | 大约 66 x 66                                   | 大约 44 x 44                                                                         | 大约 44 x 44               | 大约 44 x 44                                   | 大约 22 x 22                                 |
| 选项卡图标(可选)                       | 大约 75 x 75 (最大: 144 x 96)                    | 大约 50 x 50 (最大: 96 x 64)                                                           | 大约 50 x 50 (最大: 96 x 64) | 大约 50 x 50 (最大: 96 x 64)                     | 大约 25 x 25 (最大: 48 x 32)                   |
| App Store 报刊杂志封面图标(系统报刊杂志app显示) | 最长边至少 1024 像素                                | 最长边至少 1024 像素                                                                      | 最长边至少 1024 像素            | 最长边至少 1024 像素                                | 最长边至少 512 像素                               |
| webclip 图标(web app 和 网站应用推荐大小)  | 180 x 180                                    | 120 x 120                                                                          | 120 x 120                | 152 x 152                                    | 76 x 76                                    |

## 建议

- 所有的图像和图标建议用png格式，避免使用交错型的png文件格式；

2. 标准的图标和图像的位深（每像素占用的字节）是24位，红，绿，蓝通道分别为8位，再加上透明通道的8位，总共有32位。
3. 不需要把调色板限制在web安全色里。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-04-04 | Alfred Jiang | -  |

## 方案名称

界面设计 - 相关资料收集

## 关键字

界面设计 \ UI \ UE \ 资料收集 \ Material Design

## 需求场景

1. 界面设计参考

## 参考链接

- [GitHub - google/material-design-icons](#): Google 开源了 Material Design 系统图标包其中的 750 个字形。该系统图标包含常用的图标，如用于媒体播放、通讯、内容编辑、连接等等。在 Web 应用，安卓和 iOS 设计均适用。
- [GitHub - dekatotoro/GoogleMaterialDesignIcons](#): GoogleMaterialDesignIcons 是基于 Google Material Design Icons 的，字体文件包括了所有 material-design-icons svg 文件，非常容易使用，一共包括 1661 个图标。

## 详细内容

(见参考链接)

## 效果图

(无)

## 备注

- 文档 - iOS 人机交互指南

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-01 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

相册 - 从系统相册选择照片

## 关键字

相册 \ 拍照 \ 照片选取 \ ImagePickerController

## 需求场景

1. 需要从系统相册选择照片的一类需求

## 参考链接

1. GitHub - JSImagePickerController
2. Kayo - iOS 开发之照片框架详解

## 详细内容

使用方法：

复制 `JSImagePickerController.h/m` 文件即可，导入头文件：

```
#import "JSImagePickerController.h"
```

示例代码：

```
JSImagePickerController *imagePicker = [[JSImagePickerController alloc] init];
imagePicker.delegate = self;
[imagePicker showImagePickerInController:self animated:YES];
```

通过 `delegate` 方法获取图像：

```
- (void)imagePickerController:(UIImagePickerController *)picker didFinishPickingImage:(UIImage *)image {
    self.imageView.image = image;
}
```

多个 `delegate` 方法：

```
- (void)imagePickerControllerDidOpen;
- (void)imagePickerControllerWillOpen;
- (void)imagePickerControllerWillClose;
- (void)imagePickerControllerDidClose;
- (void)imagePickerControllerDidCancel;
```

另外以下公开属性支持自定义：

```
@property (nonatomic) NSTimeInterval animationTime;
@property (nonatomic, strong) UICollectionView *collectionView;
@property (nonatomic, strong) UIButton *photoLibraryBtn;
@property (nonatomic, strong) UIButton *cameraBtn;
@property (nonatomic, strong) UIButton *cancelBtn;
```

## 效果图

(无)

## 备注

类似推荐

- GitHub - Aevit/SCCaptureCamera



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-26 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

系统服务 - 调用系统应用和系统服务

## 关键字

系统服务 \ 相机 \ 短信 \ 录音 \ 位置服务

## 需求场景

1. 需要调用 iOS 系统应用和系统服务时

## 参考链接

1. [CocoaChina - iOS开发长文--通讯录、蓝牙、内购、GameCenter、iCloud、Passbook系统服务开发汇总](#)
2. [博客园 - iOS 8 新特性](#)

## 详细内容

### 调用系统应用

| 系统应用 | 调用方式                                                                                                      |
|------|-----------------------------------------------------------------------------------------------------------|
| 电话   | <code>tel:</code> 或者 <code>tel://</code> 、 <code>telprompt:</code> 或 <code>telprompt://</code> (拨打电话前有提示) |
| 短信   | <code>sms:</code> 或者 <code>sms://</code>                                                                  |
| 邮件   | <code>mailto:</code> 或者 <code>mailto://</code>                                                            |
| 浏览器  | <code>http:</code> 或者 <code>http://</code>                                                                |

```

// ViewController.m
// iOSSystemApplication
//
// Created by Kenshin Cui on 14/04/05.
// Copyright (c) 2014年 cmjstudio. All rights reserved.
//
#import "ViewController.h"
@interface ViewController ()  

@end
@implementation ViewController
- (void)viewDidLoad {
    [super viewDidLoad];
}
#pragma mark - UI事件
//打电话
- (IBAction)callClick:(UIButton *)sender {
    NSString *phoneNumber=@"18500138888";
//    NSString *url=[NSString stringWithFormat:@"tel://%@", phoneNumber];//这种方式会直接拨打电话
    NSString *url=[NSString stringWithFormat:@"telprompt://%@", phoneNumber];//这种方式会提示用户确认是否拨打电话
    [self openUrl:url];
}
//发送短信
- (IBAction)sendMessageClick:(UIButton *)sender {
    NSString *phoneNumber=@"18500138888";
    NSString *url=[NSString stringWithFormat:@"sms://%@", phoneNumber];
    [self openUrl:url];
}
//发送邮件
- (IBAction)sendEmailClick:(UIButton *)sender {
    NSString *mailAddress=@"kenshin@hotmail.com";
    NSString *url=[NSString stringWithFormat:@"mailto://%@", mailAddress];
    [self openUrl:url];
}
//浏览网页

```

```
- (IBAction)browserClick:(UIButton *)sender {
    NSString *url=@"http://www.cnblogs.com/kenshincui";
    [self openUrl:url];
}

#pragma mark - 私有方法
-(void)openUrl:(NSString *)urlStr{
    //注意url中包含协议名称，iOS根据协议确定调用哪个应用，例如发送邮件是"mailto://"其中"//"可以省略写成"mailto:"(其他协议也是如此)
    NSURL *url=[NSURL URLWithString:urlStr];
    UIApplication *application=[UIApplication sharedApplication];
;
    if (![application canOpenURL:url]){
        NSLog(@"无法打开%@", url, "请确保此应用已经正确安装.", url);
        return;
    }
    [[UIApplication sharedApplication] openURL:url];
}
@end
```

## 调用系统服务

| 系统服务        | <b>Framework</b>                                                              |
|-------------|-------------------------------------------------------------------------------|
| 短信与邮件       | MessageUI.framework                                                           |
| 通讯录         | AddressBook.framework & AddressBookUI.framework                               |
| 蓝牙          | GameKit.framework & MultipeerConnectivity.framework & CoreBluetooth.framework |
| 社交          | Social.framework                                                              |
| Game Center | GameKit.framework                                                             |
| 应用内购买       | StoreKit.framework                                                            |
| iCloud      | CloudKit.framework                                                            |
| Passbook    | PassKit.framework                                                             |
| 动作感应        | CoreMotion.framework                                                          |
| 定位服务 GPS    | Core Location Framework                                                       |
| 播放视频        | AVKit Framework                                                               |
| 录制视频        | AV Foundation Framework                                                       |
| 音频相关        | CoreAudioKit Framework                                                        |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-20 | Alfred Jiang | -  |
| 3  | 2016-02-14 | Alfred Jiang | -  |

## 方案名称

编码规范 - iOS 编码规范(Objective-C & Swift)

## 关键字

编码规范 \ Objective-C \ Swift

## 需求场景

- 统一团队编码风格，提高工程代码可读性与可维护性

## 参考链接

(见详细内容)

## 详细内容

### Objective-C

- [GitHub Objective-C Style Guide](#) - Style guide & coding conventions for Objective-C projects.
- [Google Objective-C Style Guide](#).
- [Macoscope Objective-C Style Guide](#) - Our code style guidelines.
- [NYTimes Objective-C Style Guide](#) - The Objective-C Style Guide used by The New York Times.
- [Robots & Pencils Objective-C Style Guide](#) - Our Objective-C coding style

guide. Fall in line!

6. [The Official raywenderlich.com Objective-C Style Guide](#) - A style guide that outlines the coding conventions for raywenderlich.com.
7. [The Official wantedly.com Objective-C Style Guide](#).
8. [Wonderful Objective-C Style Guide](#) - A complete OSX and iOS Objective-C coding style guide.
9. [GitHub - The official raywenderlich.com Objective-C style guide](#)
10. [Apple documentation - Introduction to Coding Guidelines for Cocoa](#)
11. [GitHub - Objective-C-Coding-Guidelines-In-Chinese](#)
12. [GitHub - Wordpress for iOS Style guide](#)

## Swift

1. [GitHub Swift Style Guide](#)
2. [Swift Style Guide](#)
3. [The Official raywenderlich.com Swift Style Guide](#) - The official Swift style guide for raywenderlich.com.
4. [The Swift Style Guide](#) - The Swift Style Guide. This guide should help you to improve your Swift code style, its readability, consistency and simplicity.

## 其他

1. [GitHub - Cscs](#) - A curated list of Coding Style Conventions and Standards.

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-08-05 | Alfred Jiang | -  |

## 方案名称

编码规范 - 使用 Space Commander 实现 iOS 编码规范自动化

## 关键字

编码规范 \ Space Commander \ spacecommander \ 自动化

## 需求场景

1. 统一团队编码风格，提高工程代码可读性与可维护性
2. 简化编码规范校验过程

## 参考链接

1. [JWDev - iOS 代码规范自动化](#)(推荐)
2. [GitHub - square/spacecommander](#)

## 详细内容

(见参考链接)

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-08 | Alfred Jiang | -  |

## 方案名称

网络 - iOS 访问 HTTPS SSL 和 TLS 双向加密

## 关键字

网络 \ HTTPS \ SSL \ TLS \ MKNetworkit \ AFNetworking

## 需求场景

1. 访问 HTTPS 并且使用 TLS 加密的服务器。

## 参考链接

1. 标哥的技术博客 - [iOS访问HTTPS SSL和TLS双向加密](#)
2. 简书 - [HTTPS接口加密和身份认证](#)

## 详细内容

使用 **MKNetworkit** 实现

```

- (void)testClientCertificate {
    SecIdentityRef identity = NULL;
    SecTrustRef trust = NULL;
    NSString *p12 = [[NSBundle mainBundle] pathForResource:@"testClient"
        ofType:@"p12"];
    NSData *PKCS12Data = [NSData dataWithContentsOfFile:p12];

    [[self class] extractIdentity:&identity andTrust:&trust fromPK
    CS12Data:PKCS12Data];
}

```

```

    NSString *url = @"https://218.244.131.231/ManicureShop/api/order/pay/%@";
    NSDictionary *dic = @{@"request" : @{
        @"orderNo" : @"140928210222211003064
3",
        @"type" : @(2)
    }
};

_signString = nil;
NSData *postData = [NSJSONSerialization dataWithJSONObject:dic
options:NSJ
SONWritingPrettyPrinted
error:nil
];
NSString *sign = [self signWithSignKey:@"test" params:dic];
NSMutableData *body = [postData mutableCopy];
NSLog(@"%@", [[NSString alloc] initWithData:body encoding:NSUTF8StringEncoding]);
url = [NSString stringWithFormat:url, sign];

MKNetworkEngine *engine = [[MKNetworkEngine alloc] initWithHost
Name:@"218.244.131.231"];
NSString *path = [NSString stringWithFormat:@"/ManicureShop/api/order/pay/%@", sign];
MKNetworkOperation *op = [engine operationWithPath:path params:dic httpMethod:@"POST" ssl:YES];
op.postDataEncoding = MKNKP
ostDataEncodingTypeJSON; // 传JSON

// 这个是app bundle 路径下的自签证书
op.clientCertificate = [[[NSBundle mainBundle] resourcePath]
stringByAppendingPathComponent:@"testClient.p12"];
// 这个是自签证书的密码
op.clientCertificatePassword = @"testHttps";

// 由于自签名的证书是需要忽略的，所以这里需要设置为YES，表示允许
op.shouldContinueWithInvalidCertificate = YES;
[op addCompletionHandler:^(MKNetworkOperation *completedOperat

```

```

ion) {
    NSLog(@"%@", completedOperation.responseJSON);
} errorHandler:^(MKNetworkOperation *completedOperation, NSError *error) {
    NSLog(@"%@", [error description]);
}];

[engine enqueueOperation:op];
return;
}

// 下面这段代码是提取和校验证书的数据的
+ (BOOL)extractIdentity:(SecIdentityRef *)outIdentity
    andTrust:(SecTrustRef *)outTrust
    fromPKCS12Data:(NSData *)inPKCS12Data {
OSStatus securityError = errSecSuccess;

// 证书密钥
NSDictionary *optionsDictionary = @{@"testHttps": (__bridge id )kSecImportExportPassphrase};
CFArrayRef items = CFArrayCreate(NULL, 0, 0, NULL);
securityError = SecPKCS12Import((__bridge CFDataRef)inPKCS12Data,
                               (__bridge CFDictionaryRef)optionsDictionary,
                               &items);

if (securityError == 0) {
    CFDictionaryRef myIdentityAndTrust = CFArrayGetValueAtIndex(items, 0);
    const void *tempIdentity = NULL;
    tempIdentity = CFDictionaryGetValue (myIdentityAndTrust, kSecImportItemIdentity);
    *outIdentity = (SecIdentityRef)tempIdentity;
    const void *tempTrust = NULL;
    tempTrust = CFDictionaryGetValue (myIdentityAndTrust, kSecImportItemTrust);
    *outTrust = (SecTrustRef)tempTrust;
} else {
    NSLog(@"Failed with error code %d", (int)securityError);
}
}

```

```

    return NO;
}
return YES;
}

```

使用 **AFNetworking** 实现

```

- (void)testClientCertificate {
    SecIdentityRef identity = NULL;
    SecTrustRef trust = NULL;
    NSString *p12 = [[NSBundle mainBundle] pathForResource:@"testClient"
        ofType:@"p12"];
    NSData *PKCS12Data = [NSData dataWithContentsOfFile:p12];

    [[self class] extractIdentity:&identity andTrust:&trust fromPKCS12Data:PKCS12Data];
}

NSString *url = @"https://218.244.131.231/ManicureShop/api/order/pay/%@";
NSDictionary *dic = @{@"request" : @{
        @"orderNo" : @"1409282102222110030643",
        @"type" : @(2)
    }
};

_signString = nil;
NSData *postData = [NSJSONSerialization dataWithJSONObject:dic
    options:NSJSONWritingPrettyPrinted
    error:nil];
NSString *sign = [self signWithSignKey:@"test" params:dic];
NSMutableData *body = [postData mutableCopy];
NSLog(@"%@", [[NSString alloc] initWithData:body encoding:NSUTF8StringEncoding]);
url = [NSString stringWithFormat:url, sign];

AFHTTPRequestOperationManager *manager = [AFHTTPRequestOperationManager

```

```

onManager manager];
    manager.requestSerializer = [AFJSONRequestSerializer serializer];
    manager.responseSerializer = [AFJSONResponseSerializer serializer];
    [manager.requestSerializer setValue:@"application/json" forHTTPHeaderField:@"Accept"];
    [manager.requestSerializer setValue:@"application/json" forHTTPHeaderField:@"Content-Type"];
    manager.responseSerializer.acceptableContentTypes = [NSSet setWithArray:@[@"application/json",
        @"text/plain"]];
manager.securityPolicy = [self customSecurityPolicy];

[manager POST:url parameters:dic success:^(AFHTTPRequestOperation *operation, id responseObject) {
    NSLog(@"%@", responseObject);
} failure:^(AFHTTPRequestOperation *operation, NSError *error)
{
    NSLog(@"%@", error);
}];

}

// 下面这段代码是处理SSL安全性问题的：
/* **** SSL Pinning ****/
- (AFSecurityPolicy*)customSecurityPolicy {
    NSString *cerPath = [[NSBundle mainBundle] pathForResource:@"testClient" ofType:@"cer"];
    NSData *certData = [NSData dataWithContentsOfFile:cerPath];
    AFSecurityPolicy *securityPolicy = [AFSecurityPolicy defaultPolicy];
    [securityPolicy setAllowInvalidCertificates:YES];
    [securityPolicy setPinnedCertificates:@[certData]];
    [securityPolicy setSSLPinningMode:AFSSLPinningModeCertificate];
}
/* **** SSL Pinning ****/
return securityPolicy;
}

```

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-03 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

网络 - iOS7 的多任务处理——后台获取 (Background Fetch)

## 关键字

网络 \ 多任务处理 \ 后台获取数据 \ Background Fetch \ Remote Notification

## 需求场景

- 在程序进入后台的情况下，如果你想要下载一部很大的视频以便离线观看，或者将用户图片备份到服务器时

## 参考链接

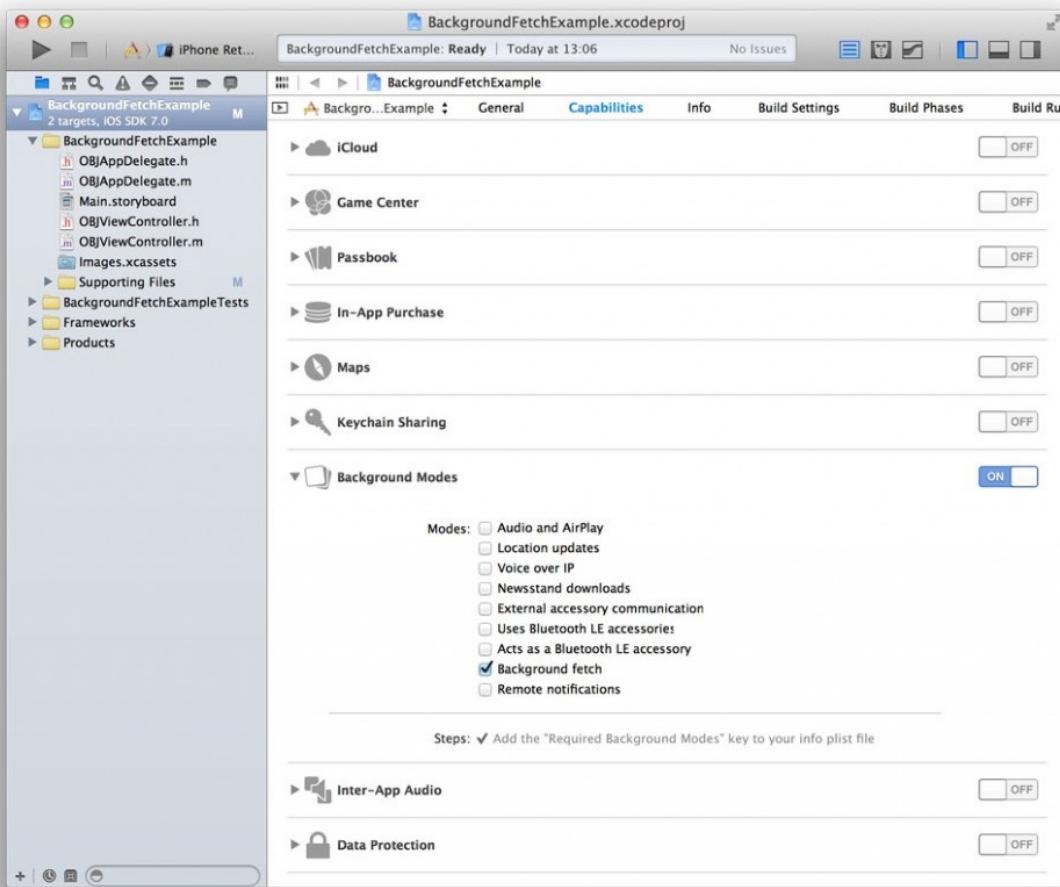
- iOS 7 系列译文 : iOS7 的多任务处理

## 详细内容

- 在 *info.plist* 文件中的 *UIBackgroundModes* 键值指定使用的特性。

```
<key>UIBackgroundModes</key>
<array>
<string>fetch</string>
</array>
```

最简单的途径是在 Xcode5 的 *project editor* 中新的性能标签页中 (*Capabilities tab*) 设置，这个标签页包含了后台模式部分，可以方便配置多任务选项。



2. 告诉 iOS 你希望多久进行一次后台获取 ``objectivec

3. (BOOL)application:(UIApplication )application

```
didFinishLaunchingWithOptions:(NSDictionary )launchOptions { [application
setMinimumBackgroundFetchInterval:UIApplicationBackgroundFetchInterval
Minimum]; return YES; } ````
```

iOS 默认不进行后台获取，所以你需要设置一个时间间隔，否则，你的应用程序永远不行在后台进行获取数据。

*UIApplicationBackgroundFetchIntervalMinimum* 这个值要求系统尽可能经常去管理应用程序什么时候会被唤醒，但如果不需要这个值，你应该指定你的时间间隔。例如，一个天气的应用程序，可能只需要几个小时才更新一次，iOS 将会在后台获取之间至少等待你指定的时间间隔。

如果你的应用允许用户退出登录，那么就没有获取新数据的需要了，你应该把 `minimumBackgroundFetchInterval` 设置为 `UIApplicationBackgroundFetchIntervalNever`，这样可以节省资源。

4. 在应用程序委托中实现下列方法````objective-c`
5. 

```
(void) application:(UIApplication )application
performFetchWithCompletionHandler:(void (^)(
(UIBackgroundFetchResult))completionHandler {
NSURLSessionConfiguration sessionConfiguration =
[NSURLSessionConfiguration defaultSessionConfiguration]; NSURLSession
session = [NSURLSession sessionWithConfiguration:sessionConfiguration];
NSURL url = [[NSURL alloc]
initWithString:@"http://yourserver.com/data.json"; NSURLSessionDataTask
task = [session dataTaskWithURL:url completionHandler:^(NSData data,
NSURLResponse response, NSError error) {

if (error) {

completionHandler(UIBackgroundFetchResultFailed);
return;
}

// Parse response/data and determine whether new content was available
BOOL hasNewData = ...

if (hasNewData) {

completionHandler(UIBackgroundFetchResultNewData);

} else {

completionHandler(UIBackgroundFetchResultNoData);

}

}];

}];
```

```
// Start the task [task resume]; } ````
```

6. 大多数情况下，无论应用在后台启动或者在前台，你会执行相同的工作，但你可以通过查看 *UIApplication* 的 *applicationState* 属性来判断应用是不是从后台启动。 ````objectivec
7. (BOOL)application:(UIApplication )application  
*didFinishLaunchingWithOptions:(NSDictionary )launchOptions {*  
NSLog(@"Launched in background %d", UIApplicationStateBackground ==  
application.applicationState); return YES; } ````
8. 测试后台获取 (Testing Background Fetch)

有两种可以模拟后台获取的途径。最简单是从 *Xcode* 运行你的应用，当应用运行时，在 *Xcode* 的 *Debug* 菜单选择 *Simulate Background Fetch* .

第二种方法，使用 *scheme* 更改 *Xcode* 运行程序的方式。在 *Xcode* 菜单的 *Product* 选项，选择 *Scheme* 然后选择 *Manage Schemes* . 在这里，你可以编辑或者添加一个新的 *scheme*，然后选中 *Launch due to a background fetch event* 。如下图：



## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-05 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

网络 - 使用 AFNetworking 实现网络请求

## 关键字

网络 \ AFNetworking \ 网络编程 \ 网络请求

## 需求场景

1. 需要实现网络交互的需求时

## 参考链接

1. [GitHub - AFNetworking](#)
2. [AFNetworking](#)
3. [AFNetworking2.0源码解析](#)

## 详细内容

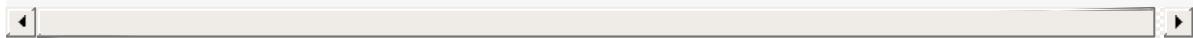
### 1. 下载 [AFNetworking](#) 并添加入工程

直接在 *Prefix.pch* 文件中引入，或者在工程的网络管理模块相关文件中引入。

### 2. 快速创建各种请求

1. [Get 请求](#)

```
AFHTTPRequestOperationManager *manager = [AFHTTPRequestOperationManager manager];
[manager GET:@"http://example.com/resources.json" parameters:nil success:^(AFHTTPRequestOperation *operation, id responseObject) {
    NSLog(@"JSON: %@", responseObject);
} failure:^(AFHTTPRequestOperation *operation, NSError *error) {
    NSLog(@"Error: %@", error);
}];
```



## 2. Post 表格数据

```
AFHTTPRequestOperationManager *manager = [AFHTTPRequestOperationManager manager];
NSDictionary *parameters = @{@"foo": @"bar"};
[manager POST:@"http://example.com/resources.json" parameters:parameters success:^(AFHTTPRequestOperation *operation, id responseObject) {
    NSLog(@"JSON: %@", responseObject);
} failure:^(AFHTTPRequestOperation *operation, NSError *error) {
    NSLog(@"Error: %@", error);
}];
```

## 3. Post 文件

```

AFHTTPRequestOperationManager *manager = [AFHTTPRequestOperationManager manager];
NSDictionary *parameters = @{@"foo": @"bar"};
NSURL *filePath = [NSURL fileURLWithPath:@"file://path/to/image.png"];
[manager POST:@"http://example.com/resources.json" parameters:parameters constructingBodyWithBlock:^(id<AFMultipartFormData> formData) {
    [formData appendPartWithFileURL:filePath name:@"image" error:nil];
} success:^(AFHTTPRequestOperation *operation, id responseObject) {
    NSLog(@"Success: %@", responseObject);
} failure:^(AFHTTPRequestOperation *operation, NSError *error) {
    NSLog(@"Error: %@", error);
}];

```

4. 创建下载任务  
`NSURLSessionConfiguration configuration = [NSURLSessionConfiguration defaultSessionConfiguration];`  
`AFURLSessionManager manager = [[AFURLSessionManager alloc] initWithSessionConfiguration:configuration];`

`NSURL URL = [NSURL URLWithString:@"http://example.com/download.zip"];`  
`NSURLRequest request = [NSURLRequest requestWithURL:URL];`  
`NSURLSessionDownloadTask downloadTask = [manager downloadTaskWithRequest:request progress:nil destination:^NSURL (NSURL *targetPath, NSURLResponse *response) { NSURL *documentsDirectoryURL = [[NSFileManager defaultManager] URLForDirectory:NSDocumentDirectory inDomain:NSUserDomainMask appropriateForURL:nil create:NO error:nil]; return [documentsDirectoryURL URLByAppendingPathComponent:[response suggestedFilename]]; } completionHandler:^(NSURLResponse *response, NSURL *filePath, NSError *error) { NSLog(@"File downloaded to: %@", filePath); }];`  
`[downloadTask resume];`

## 5. 创建上传任务

```
```objective-c
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration defaultSessionConfiguration];
AFURLSessionManager *manager = [[AFURLSessionManager alloc] initWithSessionConfiguration:configuration];

NSURL *URL = [NSURL URLWithString:@"http://example.com/upload"];
NSURLRequest *request = [NSURLRequest requestWithURL:URL];

NSURL *filePath = [NSURL fileURLWithPath:@"file://path/to/image.png"];
NSURLSessionUploadTask *uploadTask = [manager uploadTaskWithRequest:request fromFile:filePath progress:nil completionHandler:^(NSURLResponse *response, id responseObject, NSError *error) {
    if (error) {
        NSLog(@"Error: %@", error);
    } else {
        NSLog(@"Success: %@ %@", response, responseObject);
    }
}];
[uploadTask resume];
```

## 1. 创建多文件上传，并显示进度

```
```objective-c
NSMutableURLRequest *request
= [[AFHTTPRequestSerializer serializer]
multipartFormRequestWithMethod:@"POST"
URLString:@"http://example.com/upload" parameters:nil
constructingBodyWithBlock:^(id formData) {

    [formData appendPartWithFileURL:[NSURL fileURLWithPath:@"file://path/to/image.jpg"] name:@"file" fileName:@"filename.jpg" mimeType:@"image/jpeg" error:nil];
}

} error:nil];
```

```
AFURLSessionManager *manager = [[AFURLSessionManager alloc]
initWithSessionConfiguration:[NSURLSessionConfiguration
defaultSessionConfiguration]]; NSProgress *progress = nil;

NSURLSessionUploadTask *uploadTask = [manager
uploadTaskWithStreamedRequest:request progress:&progress
completionHandler:^(NSURLResponse *response, id responseObject, NSError
*error) { if (error) { NSLog(@"Error: %@", error); } else { NSLog(@"%@", response, responseObject); } }];
[uploadTask resume];
```

#### 7. 创建一个 Data 的上传下载任务

```
```objective-c
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration
defaultSessionConfiguration];
AFURLSessionManager *manager = [[AFURLSessionManager alloc] init
withSessionConfiguration:configuration];

NSURL *URL = [NSURL URLWithString:@"http://example.com/upload"];
NSURLRequest *request = [NSURLRequest requestWithURL:URL];

NSURLSessionDataTask *dataTask = [manager dataTaskWithRequest:re
quest completionHandler:^(NSURLResponse *response, id responseObject,
NSError *error) {
    if (error) {
        NSLog(@"Error: %@", error);
    } else {
        NSLog(@"%@", response, responseObject);
    }
}];
[dataTask resume];
```

### 3. 网络状态监测

```
[[AFNetworkReachabilityManager sharedManager] setReachabilityStatusChangeBlock:^(AFNetworkReachabilityStatus status) {
    NSLog(@"Reachability: %@", AFStringFromNetworkReachabilityStatus(status));
}];
```

#### 4. 对 UIKit 的功能扩展

##### UIKit+AFNetworking.h

```
#import <UIKit/UIKit.h>

#ifndef _UIKIT_AFNETWORKING_
#define _UIKIT_AFNETWORKING_

#import "AFNetworkActivityIndicatorManager.h"

#import "UIActivityIndicatorView+AFNetworking.h"
#import "UIAlertView+AFNetworking.h"
#import "UIButton+AFNetworking.h"
#import "UIImageView+AFNetworking.h"
#import "UIKit+AFNetworking.h"
#import "UIProgressView+AFNetworking.h"
#import "UIWebView+AFNetworking.h"
#endif /* _UIKIT_AFNETWORKING_ */
```

#### 效果图

(无)

#### 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-08-18 | Alfred Jiang | -  |

## 方案名称

网络 - 使用 CTTelephonyNetworkInfo 确定移动网络类型和运营商信息

## 关键字

网络 \ CTTelephonyNetworkInfo \ 移动网络类型 \ 运营商信息

## 需求场景

1. 需要确认手机移动网络状态和运营商名称等信息

## 参考链接

1. Stack Overflow - CTTelephonyNetworkInfo detect data type(推荐)
2. Stack Overflow - Detect carrier connection type (3G / EDGE / GPRS)

## 详细内容

### 方案一

CTTelephonyNetworkInfo+CellularConnectionName.h

```
#import <CoreTelephony/CTTelephonyNetworkInfo.h>
#import <Foundation/Foundation.h>

@interface CTTelephonyNetworkInfo (CellularConnectionName)

- (NSString *)cellularConnectionVersionName;
- (NSString *)cellularConnectionNiceName;

@end
```

## CTTelephonyNetworkInfo+CellularConnectionName.m

```
#import "CTTelephonyNetworkInfo+CellularConnectionName.h"

@implementation CTTelephonyNetworkInfo (CellularConnectionName)

- (NSString *)cellularConnectionVersionName {

    if ([self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYGPRS]
        || [self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYEDGE]) {

        return @"2G";
    }

    else if ([self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYWCDMA]
             || [self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYHSDPA]
             || [self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYHSUPA]
             || [self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYCDMA1X]
             || [self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYCDMAEVDOREV0]
             || [self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYCDMAEVDOREV A]
             || [self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYCDMAEVDOREV B]) {

        return @"3G";
    }

    else if ([self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYLTE]
             || [self.currentRadioAccessTechnology isEqualToString:CTRADIOACCESSTECHNOLOGYHSPA])
        return @"4G";
    }

    return @"Unknown";
}
```

```
        ||[self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyHRPD]) {

            return @"3G";
        }
        else if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyLTE]) {

            return @"4G";
        }

        return @"UNKNOWN";
    }

- (NSString *)cellularConnectionNiceName {

    if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyGPRS]) {
        return @"GPRS";
    } else if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyEdge]) {
        return @"EDGE";
    } else if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyWCDMA]) {
        return @"WCDMA";
    } else if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyHSDPA]) {
        return @"HSDPA";
    } else if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyHSUPA]) {
        return @"HSUPA";
    } else if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyCDMA1x]) {
        return @"CDMA1X";
    } else if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyCDMAEVDORev0]) {
        return @"CDMAEVDOREV0";
    } else if ([self.currentRadioAccessTechnology isEqualToString:CTRadioAccessTechnologyCDMAEVDORevA]) {
        return @"CDMAEVDOREVA";
    }
}
```

```

    } else if ([self.currentRadioAccessTechnology isEqualToString:@"CDMAEVDORevB"]) {
        return @"CDMAEVDOREVB";
    } else if ([self.currentRadioAccessTechnology isEqualToString:@"CTRadioAccessTechnologyeHRPD"]) {
        return @"EHRPD";
    } else if ([self.currentRadioAccessTechnology isEqualToString:@"CTRadioAccessTechnologyLTE"]) {
        return @"LTE";
    }

    return @"UNKNOWN";
}

@end

```

使用

```

CTTelephonyNetworkInfo *telephonyInfo = [CTTelephonyNetworkInfo new];
NSLog(@"Current Radio Access Technology: %@", telephonyInfo.currentRadioAccessTechnology);
[NSNotificationCenter.defaultCenter addObserverForName:CTRadioAccessTechnologyDidChangeNotification
                                         object:nil
                                           queue:nil
                                         usingBlock:^(NSNotification *note)
{
    NSLog(@"New cellularConnectionVersionName: %@", telephonyInfo.cellularConnectionVersionName);
    NSLog(@"New cellularConnectionNiceName: %@", telephonyInfo.cellularConnectionNiceName);
}];

```

方案二

CheckNetwork.h

```

// CheckNetwork.h
// CheckNetworkType
//
// Created by viktyz on 16/10/27.
// Copyright © 2016年 AlfredJiang. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface CheckNetwork : NSObject

+ (NSString *)networkType;

@end

```

## CheckNetwork.m

```

// CheckNetwork.m
// CheckNetworkType
//
// Created by viktyz on 16/10/27.
// Copyright © 2016年 AlfredJiang. All rights reserved.
//


#import "CheckNetwork.h"
#import <netinet/in.h>
#import <SystemConfiguration/SCNetworkReachability.h>

@implementation CheckNetwork

+ (NSString *)networkType
{
    struct sockaddr_in zeroAddress;
    bzero(&zeroAddress, sizeof(zeroAddress));
    zeroAddress.sin_len = sizeof(zeroAddress);
    zeroAddress.sin_family = AF_INET;
    SCNetworkReachabilityRef defaultRouteReachability = SCNetwor

```

```

kReachabilityCreateWithAddress(NULL, (struct sockaddr *)&zeroAddress);
    SCNetworkReachabilityFlags flags;
    SCNetworkReachabilityGetFlags(defaultRouteReachability, &flags);

    if ((flags & kSCNetworkReachabilityFlagsReachable) == 0)
    {
        return @"Not Reachable";
    }

    if (((flags & kSCNetworkReachabilityFlagsConnectionOnDemand
) != 0) ||
        (flags & kSCNetworkReachabilityFlagsConnectionOnTraffic
) != 0))
    {
        if ((flags & kSCNetworkReachabilityFlagsInterventionRequ
ired) == 0)
        {
            return @"wifi";
        }
    }

    if((flags & kSCNetworkReachabilityFlagsReachable) == kSCNetw
orkReachabilityFlagsReachable){

        if ((flags & kSCNetworkReachabilityFlagsIsWWAN) == kSCNe
tworkReachabilityFlagsIsWWAN){

            if ((flags & kSCNetworkReachabilityFlagsTransientCon
nection) == kSCNetworkReachabilityFlagsTransientConnection){

                if((flags & kSCNetworkReachabilityFlagsConnectio
nRequired) == kSCNetworkReachabilityFlagsConnectionRequired){

                    return @"2g";
                }
                return @"3g";
            }
            return @"4g";
        }
    }
}

```

```
        }

    }

    if ((flags & kSCNetworkReachabilityFlagsConnectionRequired) == 0)
    {
        return @"wifi";
    }

    return @"Unknown";
}

@end
```

使用

```
[CheckNetwork networkType];
```

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-06-12 | Alfred Jiang | -  |

## 方案名称

网络 - 使用 SimplePingHelper 在 iPhone / iPad 上 ping 指定服务器

## 关键字

网络 \ ping \ 检测服务器连接

## 需求场景

1. 测试 ping 连接

## 参考链接

1. Chris Hulbert - How to ping a server in Objective-C / iPhone(推荐)
2. GitHub - chrishulbert/SimplePingHelper

## 详细内容

1. 将 **SimplePingHelper** 相关代码引入工程

2. 使用方法如下

```
- (void)tapPing {
    [SimplePingHelper ping:@"www.google.com"
        target:self sel:@selector(pingResult:)];
}

- (void)pingResult:(NSNumber*)success {
    if (success.boolValue) {
        [self log:@"SUCCESS"];
    } else {
        [self log:@"FAILURE"];
    }
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-18 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

网络 - 判断连接状态

## 关键字

网络 \ 网络连接 \ Network \ Connection

## 需求场景

1. 需要对网络状态进行判断的场景

## 参考链接

(无)

## 详细内容

1. 导入 **SystemConfiguration.framework**, 并 **#import**

2. 判断设备是否联网

```
+ (BOOL)connectedToNetwork{
    // 创建零地址，0.0.0.0的地址表示查询本机的网络连接状态

    struct sockaddr_storage zeroAddress;
    bzero(&zeroAddress, sizeof(zeroAddress));
    zeroAddress.ss_len = sizeof(zeroAddress);
    zeroAddress.ss_family = AF_INET;

    // Recover reachability flags
    SCNetworkReachabilityRef defaultRouteReachability = SCNetworkReachabilityCreateWithAddress(NULL, (struct sockaddr *)&zeroAddress);
    SCNetworkReachabilityFlags flags;

    // 获得连接的标志
    BOOL didRetrieveFlags = SCNetworkReachabilityGetFlags(defaultRouteReachability, &flags);
    CFRelease(defaultRouteReachability);

    // 如果不能获取连接标志，则不能连接网络，直接返回
    if (!didRetrieveFlags)
    {
        return NO;
    }
    // 根据获得的连接标志进行判断

    BOOL isReachable = flags & kSCNetworkFlagsReachable;
    BOOL needsConnection = flags & kSCNetworkFlagsConnectionRequired;
    return (isReachable&&!needsConnection) ? YES : NO;
}
```

## 效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-03 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

设计模式 - iOS 开发常用设计模式简介

## 关键字

设计模式 \ Design Patterns

## 需求场景

1. 帮助了解 iOS App 开发常用设计模式

## 参考链接

1. [ITEYE - IOS 设计模式之一（MVC模式，单例模式）\(推荐译文\)](#)
2. [Ray Wenderlich - iOS Design Patterns](#)
3. [GitHub - Design-Patterns-In-Swift](#)

## 详细内容

一、创建型：

1. **单例模式 (The Singleton Pattern)**

各种 `sharedInstanceManager`

二、结构型：

1. **模型-视图-控制器 (MVC) 以及 MVVC**

## 各种 UIViewController

### 2. 装饰器 (The Decorator Design Pattern)

#### Category(类别) 和 Delegation (委托)

装饰器模式在不修改原来代码的情况下动态的给对象增加新的行为和职责，它通过一个对象包装被装饰对象的方法来修改类的行为，这种方法可以做为子类化的一种替代方法。

在 Objective-C 中，存在两种非常常见的实现: Category(类别) 和 Delegation (委托)。

### 3. 适配器模式 (The Adapter Pattern)

适配器可以让一些接口不兼容的类一起工作。它包装一个对象然后暴露一个标准的交互接口。如果你熟悉适配器设计模式，苹果通过一个稍微不同的方式来实现它-苹果使用了协议的方式来实现。你可能已经熟悉 UITableViewDelegate , UIScrollViewDelegate , NSCoder 和 NSCopying 协议。举个例子，使用 NSCopying 协议，任何类都可以提供一个标准的 copy 方法。

### 4. 外观 (门面) (The Facade Design Pattern) :

#### 各种接口类

门面模式针对复杂的子系统提供了单一的接口，不需要暴露一些列的类和API给用户，你仅仅暴露一个简单统一的API。

### 三、行为型：

#### 1. 观察者模式 (The Observer Pattern)

##### 苹果的推送通知 (Push Notification)

##### 通知 (Notifications)

##### Key-Value Observing(KVO)

#### 2. 备忘录模式 (The Memento Pattern)

##### 归档 (Archiving)

备忘录模式快照对象的内部状态并将其保存到外部。换句话说，它将状态保存到某处，过会你可以不破坏封装的情况下恢复对象的状态，也就是说原来对象中的私有数据仍然是私有的。

```

- (void)saveCurrentState
{
    // When the user leaves the app and then comes back again, he wants it to be in the exact same state
    // he left it. In order to do this we need to save the currently displayed album.
    // Since it's only one piece of information we can use NSUserDefaults.
    [[NSUserDefaults standardUserDefaults] setInteger:currentAlbumIndex forKey:@"currentAlbumIndex"];
}

- (void)loadPreviousState
{
    currentAlbumIndex = [[NSUserDefaults standardUserDefaults] integerForKey:@"currentAlbumIndex"];
    [self showDataForAlbumAtIndex:currentAlbumIndex];
}

```

### 3. 命令模式 (The Command Pattern)

苹果通过 Target-Action 机制和 Invocation 实现命令模式。

命令模式将一个请求封装为一个对象。封装以后的请求会比原生的请求更加灵活，因为这些封装后的请求可以在多个对象之间传递，存储以便以后使用，还可以动态的修改，或者放进一个队列中

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-03 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

设计模式 - 使用命令模式实现撤销删除

## 关键字

设计模式 \ 命令模式 \ 撤销删除

## 需求场景

- 较为复杂的撤销删除需求

## 参考链接

- IOS设计模式之四（备忘录模式，命令模式）

## 详细内容

命令模式将一个请求封装为一个对象。封装以后的请求会比原生的请求更加灵活，因为这些封装后的请求可以在多个对象之间传递，存储以便以后使用，还可以动态的修改，或者放进一个队列中。苹果通过 *Target-Action* 机制和 *Invocation* 实现命令模式。

```
- (void)deleteAlbum
{
    // 1
    Album *deletedAlbum = allAlbums[currentAlbumIndex];

    // 2
    NSMethodSignature *sig = [self methodSignatureForSelector:@selector(addAlbumAtIndex:)];
    NSInvocation *undoAction = [NSInvocation invocationWithMethodSignature:	sig];
    [undoAction setTarget:self];
    [undoAction setSelector:@selector(addAlbumAtIndex:)];
    [undoAction setArgument:&deletedAlbum atIndex:2];
    [undoAction setArgument:&currentAlbumIndex atIndex:3];
    [undoAction retainArguments];

    // 3
    [undoStack addObject:undoAction];

    // 4
    [[LibraryAPI sharedInstance] deleteAlbumAtIndex:currentAlbumIndex];
    [self reloadScroller];

    // 5
    [toolbar.items[0] setEnabled:YES];
}
```

上面的代码中有一些新的激动人心的特性，所以下面我们就来考虑每个被标注了注释的地方：

1. 获取需要删除的专辑

2. 定义了一个类型为*NSMethodSignature*的对象去创建*NSInvocation*, 它将用来撤销删除操作。*NSInvocation*需要知道三件事情：选择器（发送什么消息），目标对象（发送消息的对象），还有就是消息所需要的参数。在上面的例子中，消息是与删除方法相反的操作，因为当你想撤销删除的时候，你需要将刚删除的数据回加回去。

3. 创建了*undoAction*以后，你需要将其增加到*undoStack*中。撤销操作将被增加在数组的末尾。

4. 使用*LibraryAPI*删除专辑，然后重新加载滚动视图。

5. 因为在撤销栈中已经有了操作，你需要使得撤销按钮可用。

注意：使用 *NSInvocation*，你需要记住下面的几点：

1. 参数必须以指针的形式传递。

2. 参数从索引2开始，索引0，1为目标（*target*）和选择器（*selector*）保留。

3. 如果参数有可能会被销毁，你需要调用*retainArguments*.

撤销方法：

```
- (void)undoAction
{
    if (undoStack.count > 0)
    {
        NSInvocation *undoAction = [undoStack lastObject];
        [undoStack removeLastObject];
        [undoAction invoke];
    }

    if (undoStack.count == 0)
    {
        [toolbar.items[0] setEnabled:NO];
    }
}
```

## 效果图

(无)

## 备注

更多设计模式介绍请参考专题

- [设计模式 - iOS 开发常用设计模式简介](#)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

设计模式 - 单例模式

## 关键字

设计模式 \ 单例 \ GCD \ dispatch\_once \ 设计模式

## 需求场景

1. 单例场景，如某个全局使用的管理类

## 参考链接

(无)

## 详细内容

1. Swift 解决方案

```
class var sharedInstance : SettingManager {  
    struct Static {  
        static var onceToken : dispatch_once_t = 0  
        static var instance : SettingManager? = nil  
    }  
    dispatch_once(&Static.onceToken) {  
        Static.instance = SettingManager()  
    }  
    return Static.instance!  
}
```

## 2. Objective-C 解决方案

```
+ (DBManager *)sharedManager  
{  
    static DBManager *sharedManager;  
  
    static dispatch_once_t onceToken;  
    dispatch_once(&onceToken, ^{  
        sharedManager = [[DBManager alloc] init];  
    });  
  
    return sharedManager;  
}
```

## 效果图

(无)

## 备注

- 设计模式 - iOS 开发常用设计模式简介

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-26 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

语法 - ID类字段生成实现

## 关键字

语法 \ 唯一ID \ ID \ 不重复ID

## 需求场景

1. 需要生成不重复的ID类字段时

## 参考链接

(无)

## 详细内容

```
+ (NSString *)randomId
{
    NSString *strId = [[NSString stringWithFormat:@"%f", [[NSDate
date] timeIntervalSince1970]] stringByReplacingOccurrencesOfString:@"."
 withString:@""];
    return [strId stringByAppendingString:[NSString stringWithFormat:@"%u",
arc4random_uniform(10000)]];
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-13 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

语法 - If not let - in Swift

## 关键字

语法 \ optional \ ? \ ! \ if let

## 需求场景

1. 判断 optional 字段是否有效\无效时

## 参考链接

1. Stack Overflow - If not let in Swift

## 详细内容

正常的 *if let* 用法

```
if let type = json.type {  
  
} else {  
    //There is no type in the root element  
}
```

*if not let* 实现

```
func ifNotLet<T>(optional: T?, closure: @autoclosure () -> ()) -> T {
    switch optional {
        case .None:
            closure()
        case let .Some(value)
            return value
    }
}

let type = ifNotLet(json.type) {
    XCTFail("There is no type in the root element")
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-22 | Alfred Jiang | -  |

## 方案名称

语法 - iOS 消息传递机制

## 关键字

语法 \ 消息传递机制

## 需求场景

- 熟悉 iOS 消息传递机制

## 参考链接

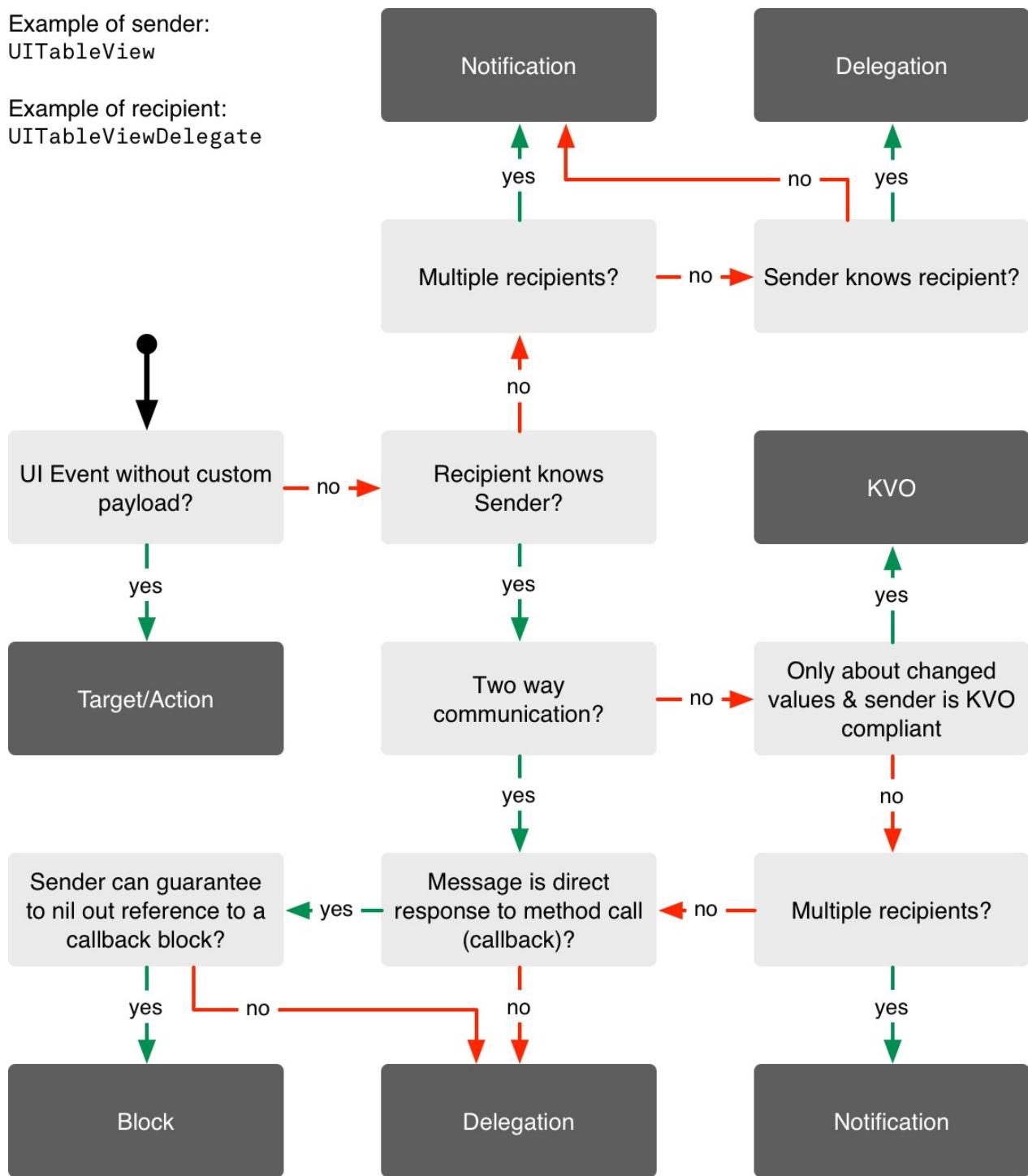
- [bjccn - 消息传递机制](#)
- [Objc - Communication Patterns](#)

## 详细内容

(见参考链接)

Example of sender:  
UITableView

Example of recipient:  
UITableViewDelegate



## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-20 | Alfred Jiang | -  |

## 方案名称

语法 - MRC 下全局变量的 Delegate 陷阱

## 关键字

语法 \ Objective-C \ MRC \ 成员变量 \ Delegate

## 需求场景

1. 使用全局变量时赋值 delegate 需要注意的地方

## 参考链接

(无)

## 详细内容

在 MRC 代码中，当我们对全局成员变量的 delegate 进行赋值时，一定要记得在 dealloc 中释放该全局变量的同时对其中的 delegate 进行置 nil 操作。

否则会在下面的代码中造成 crash。

```
if (delegate && [delegate respondsToSelector:@selector(anymethod
:)]) {
    [delegate setAnymethod:nil];
}
```

原因是 MRC 中 delegate 为 assign 修饰，assign 修饰在 self 置 nil 时该变量并不会置 nil。

所以我们必须在 dealloc 中显式的对 delegate 进行置 nil 操作。

与之对应的 ARC 中 delegate 为 weak 修饰，weak 修饰在 self 置 nil 时会自动对 delegate 进行置 nil 操作。

这也算是 ARC 的一个进步之处吧，缺点是 weak 的对象必须是 Objective-C 的对象。而 assign 可以修饰简单数据类型。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-05-12 | Alfred Jiang | -  |

## 方案名称

语法 - NSString \ NSData \ NSArray \ NSDictionary 格式转换 ( NSArray / NSDictionary <=> NSData <=> NSString )

## 关键字

语法 \ 数据存储 \ 格式转换 \ NSString \ NSData \ NSArray \ NSDictionary \ NSArray / NSDictionary <=> NSData <=> NSString

## 需求场景

1. 进行 NSString \ NSData \ NSArray \ NSDictionary 格式转换

## 参考链接

1. Sina - NSJSONSerialization-JSON数据与NSDictionary和NSArray之间的转化

## 详细内容

### 1. NSArray / NSDictionary <=> NSData ( Json )

```
// 将字典或者数组转化为JSON串
- (NSData *)toJSONData:(id)data{
    if (!data) {
        NSLog(@"Please Check %s parameter", __FUNCTION__);
        return nil;
    }
}
```

```
NSError *error = nil;
NSData *jsonData = [NSJSONSerialization dataWithJSONObject:dict
options:NSJSONWritingPrettyPrinted
error:&error];

if ([jsonData length] != 0 && error == nil){
    return jsonData;
}
else{
    return nil;
}

// 将JSON串转化为字典或者数组
- (id)toArrayOrDictionary:(NSData *)jsonData{

    if (!jsonData || [jsonData length] == 0) {
        NSLog(@"Please Check %s parameter", __FUNCTION__);
        return nil;
    }

    NSError *error = nil;
    id jsonObject = [NSJSONSerialization JSONObjectWithData:jsonData
options:NSJSONReadingAllowFragments
error:&error];
}

if (jsonObject != nil && error == nil){
    return jsonObject;
}
```

语法 - NSString \ NSData \ NSArray \ NSDictionary 格式转换 ( NSArray / NSDictionary <=> NSData <=> NSString )

```
else{  
    return nil;  
}  
}
```

### 3. NSData <=> NSString ( Json )

```
NSData *jsonData = [jsonString dataUsingEncoding:NSUTFStringEncoding];  
  
NSString *jsonString = [[NSString alloc] initWithData:jsonData encoding:NSUTFStringEncoding];
```

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-22 | Alfred Jiang | -  |

## 方案名称

语法 - Objective-C Runtime 介绍

## 关键字

语法 \ Runtime

## 需求场景

- 熟悉 Objective-C Runtime

## 参考链接

- [Yulingtianxia's Blog - Objective-C Runtime](#)(推荐)

## 详细内容

(见参考链接) : 主要内容如下

引言

简介

与 **Runtime** 交互

**Runtime** 术语

消息

动态方法解析

消息转发

**健壮的实例变量(Non Fragile ivars)**

**Objective-C Associated Objects**

**Method Swizzling**

总结

**效果图**

(无)

**备注**

- Apple documentation - Objective-C Runtime Programming Guide
- Apple documentation - Objective-C Runtime Reference
- 特酷吧 - Objective-C Runtime 分析

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-08 | Alfred Jiang | -  |

## 方案名称

语法 - Objective-C 与 Swift 互相调用

## 关键字

语法 \ Objective-C \ Swift

## 需求场景

1. swift、Objective-C 混编

## 参考链接

1. [KittenYang - 在Swift项目中使用OC，在OC项目中使用Swift](#)

## 详细内容

### Objective-C 调用 Swift

方法:

1. 在工程的 *Build Settings* 中把 *defines module* 设为 YES 。
2. 把 *product module name* 设置为项目工程的名字。
3. 在需要调用 Swift 的 Objective-C 文件中导入 *ProjectName-Swift.h* 。

注意:

1. 调用 Swift 类的话，需要在 Swift 的 class 前面加 @objc 修饰。
2. 在 Objective-C 工程中首次创建 Swift 文件时，系统会自动帮你创建 Objective-

*C Bridging Header*。该文件主要用于 *Swift* 引用 *Objective-C* 代码。

## Swift 调用 Objective-C

方法：

1. 创建 *Objective-C Bridging Header* 文件，命名为 *ProjectName-Bridging-Header.h*。
2. 在 *ProjectName-Bridging-Header.h* 中添加需要被 *Swift* 调用的 *Objective-C* 头文件。
3. 在工程的 *Build Settings* 中把 *Objective-C Bridging Header* 路径设置为 *ProjectName-Bridging-Header.h* 路径（路径必须指向文件本身）。

注意：

1. 在 *Swift* 工程中首次创建 *Objective-C* 文件时，系统会自动帮你创建 *Objective-C Bridging Header*。选择 YES 会自动设置好 *Objective-C Bridging Header* 路径。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-20 | Alfred Jiang | -  |

## 方案名称

语法 - Objective-C 中 BOOL 类成员变量的判断陷阱

## 关键字

语法 \ Objective-C \ BOOL \ 成员变量 \ 判断

## 需求场景

- 对某对象的 BOOL 类成员变量做肯定判断时，可能出现的判断错误

## 参考链接

(无)

## 详细内容

当自定义的类中包含 BOOL 类公共成员变量时，若该类的对象为空，

对该对象 BOOL 类成员变量做肯定判断时，无论逻辑上 BOOL 类成员变量是否为 YES，都会判断为 NO。

```
TestObject *tObject0 = nil;  
  
tObject0.isEmpty = YES;  
  
if (tObject0.isEmpty) {  
    NSLog(@"tObject0 is Empty");  
}  
else  
{  
    NSLog(@"tObject0 is not Empty");  
}  
  
TestObject *tObject1 = nil;  
  
tObject1.isEmpty = NO;  
  
if (tObject1.isEmpty) {  
    NSLog(@"!tObject1 is Empty");  
}  
else  
{  
    NSLog(@"tObject1 is not Empty");  
}
```

输出：

```
2016-01-20 12:53:04.527 TestEmpty[11315:335458] tObject0 is not  
Empty  
2016-01-20 12:53:04.528 TestEmpty[11315:335458] tObject0 is not  
Empty
```

对该对象 BOOL 类成员变量做否定判断时，无论逻辑上 BOOL 类成员变量是否为 NO，都会判断为 YES。

```

TestObject *tObject0 = nil;

tObject0.isEmpty = YES;

if (!tObject0.isEmpty) {
    NSLog(@"tObject0 is Empty");
}
else
{
    NSLog(@"tObject0 is not Empty");
}

TestObject *tObject1 = nil;

tObject1.isEmpty = NO;

if (!tObject1.isEmpty) {
    NSLog(@"!tObject1 is Empty");
}
else
{
    NSLog(@"tObject1 is not Empty");
}

```

输出：

```

2016-01-20 12:54:48.837 TestEmpty[11330:337224] tObject0 is Empt
y
2016-01-20 12:54:48.838 TestEmpty[11330:337224] !tObject0 is Emp
ty

```

所以在针对此类成员变量判断时，一定要优先判断对象是否为空，只有在对象非空的条件下，判断 BOOL 类成员变量才有效。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-22 | Alfred Jiang | -  |

## 方案名称

语法 - Objective-C 中的 Meta-class 是什么

## 关键字

语法 \ Meta-class \ 元类

## 需求场景

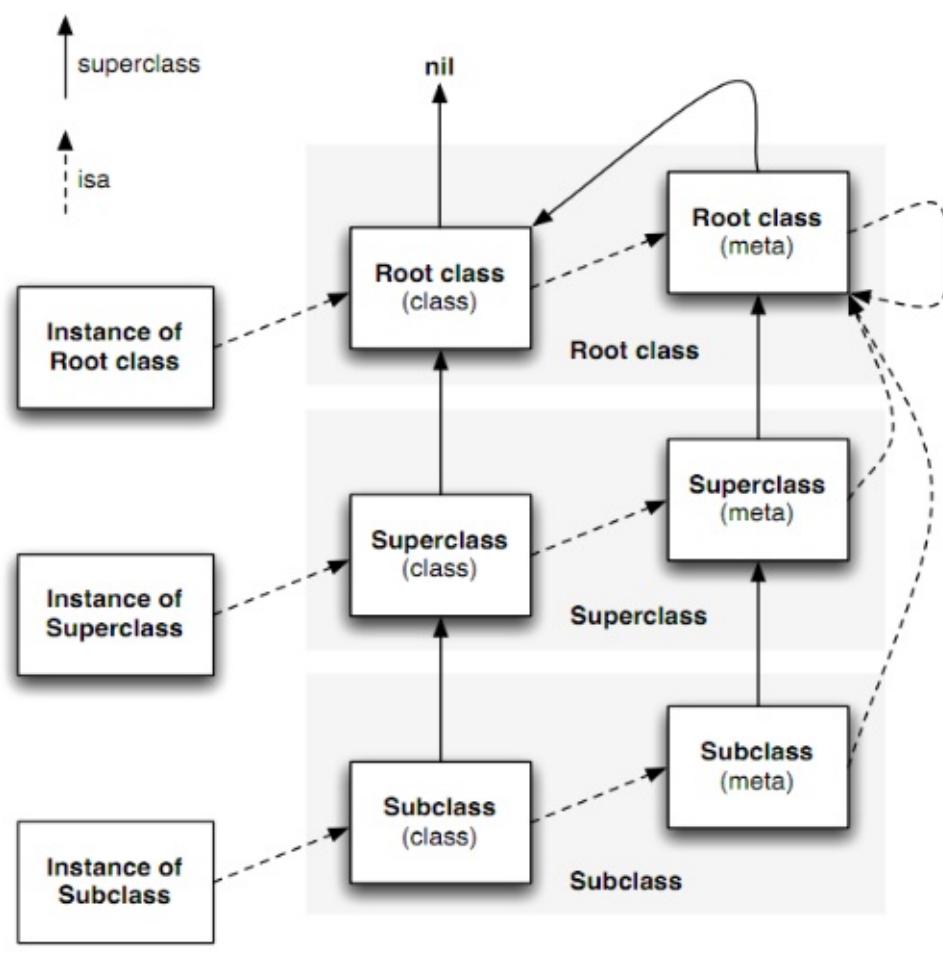
- 熟悉 Objective-C 类结构

## 参考链接

- 伯乐在线 - Objective-C 中的 Meta-class 是什么
- Cocoa with Love - What is a meta-class in Objective-C?

## 详细内容

(见参考链接)



## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-22 | Alfred Jiang | -  |

## 方案名称

语法 - Objective-C 基础集合类介绍

## 关键字

语法 \ 基础集合类

## 需求场景

- 熟悉 Objective-C 基础集合类

## 参考链接

- [CocoaChina - 基础集合类](#)
- [Objc - The Foundation Collection Classes](#)

## 详细内容

(见参考链接) : 主要介绍以下基础集合类

### **NSArray**

### **NSDictionary**

### **NSSet**

### **NSOrderedSet**

### **NSMutableDictionary**

**NSMapTable**

**NSPointerArray**

**NSCache**

**NSIndexSet**

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-03 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |
| 2  | 2016-02-01 | Alfred Jiang | -  |

## 方案名称

语法 - 使用字面量

## 关键字

语法 \ 字面量 \ 字面值

## 需求场景

1. 提高代码可读性

## 参考链接

1. [IOS使用字面值](#)
2. [Swifter - 字面量转换](#)
3. [GitHub - Literally](#)
4. [Clang 3.9 documentation - Objective-C Literals](#)

## 详细内容

### 1. Swift 解决方案

#### NSString

```
let aString = "Hello"
```

## Number

```
let aNumber : NSNumber = 3
let aBool : NSNumber = true
```

## NSArray

```
let anArray = [1, 2, 3]
```

## Dictionary

```
let aDictionary = ["key1": "value1", "key2": "value2"]
```

Swift 为我们提供了一组非常有意思的接口，用来将字面量转换为特定的类型。对于那些实现了字面量转换接口的类型，在提供字面量赋值的时候，就可以简单地按照接口方法中定义的规则“无缝对应”地转换为这些类型。这些接口包括了各个原生的字面量，它们是：

ArrayLiteralConvertible

BooleanLiteralConvertible

CharacterLiteralConvertible

DictionaryLiteralConvertible

ExtendedGraphemeClusterLiteralConvertible

FloatLiteralConvertible

NilLiteralConvertible

IntegerLiteralConvertible

StringLiteralConvertible

StringInterpolationConvertible

用法举例：

通常通过 NSString 定义一个 NSURL 的方法如下

```
let url = NSURL(string: "http://swifter.tips")
```

使用 `StringLiteralConvertible` 构造一个 `NSURL` 的 extension

```
extension NSURL: StringLiteralConvertible {
    public class func
        convertFromStringLiteral(value: String) -> Self
    {
        return self(string: value)
    }

    public class func
        convertFromExtendedGraphemeClusterLiteral(value: String)
-> Self
    {
        return self(string: value)
    }
}
```

那么我们现在可以直接使用如下方式直接赋值 `NSString` 生成 `NSURL`

```
let url: NSURL = "http://swifter.tips"
```

## 2. Objective-C 解决方案

### NSString

```
NSString *str = [[NSString alloc] initWithFormat:@"%@", @"Hello World"];
//---->字面值如下

NSString *str = @"Hello World";
```

### Number

```
NSNumber *someNumber = [NSNumber numberWithInt:1];
```

//---->字面值如下

```
NSNumber *intNumber = @1;
NSNumber *floatNumber = @2.5f;
NSNumber *doubleNumber = @3.14159;
NSNumber *boolNumber = @YES;
NSNumber *charNumber = @'a';
```

## NSArray

```

NSArray *animals =[NSArray arrayWithObjects:@"cat", @"dog", @"mouse",
                     @"badger", nil];

//---->字面值如下

NSArray *animals = @[@"cat", @"dog", @"mouse", @"badger"];

NSString *dog = [animals objectAtIndex:1];

//---->通过字面值取数据

NSString *dog = animals[1];

//可变的话可以直接设置新值
[mutableArray replaceObjectAtIndex:1 withObject:@"dog"];

//---->字面值设置值

mutableArray[1] = @"dog";


NSString *str1;
NSString *str2 = @"1";
NSString *str3 = @"1";

//插入空数据会抛出异常
NSArray *arr = @[str1,str2,str3];

//reason: '*** -[__NSPlaceholderArray initWithObjects:count:]:
//attempt to insert nil object from objects[0]'

//这种方式不会抛异常，但是数组会收到第一个nil就终止了。所以采用字面值形式会
//得到错误提示。
NSArray *arr = [NSArray arrayWithObjects:str1,str2,str3, nil];
NSLog(@"%@",arr);
NSLog(@"%@",arr.firstObject);

```

## Dictionary

```
NSDictionary *personData =[NSDictionary dictionaryWithObjectsAnd
Keys:@"Kevin", @"firstName",@"Jin", @"lastName",[NSNumber numberWithInt:25], @"age", nil];

//---->字面值如下

NSDictionary *personData = @{@"firstName" : @"Kevin",@"lastName"
: @"Jin", @"age" : @25};

//----->字面值

NSString *str1;
NSString *str2 = @"1";
NSString *str3 = @"1";
//和数组一样，value不能有nil，否则就会抛出异常。
NSDictionary *dic = @{@"str1":str1,@"str2":str2,@"str3":str3};

//reason: '*** -[__NSPlaceholderDictionary initWithObjects:forKe
ys:count:]: attempt to insert nil object from objects[0]'

//不会提示你错误
NSDictionary *dic = [NSDictionary dictionaryWithObjectsAndKeys:s
tr1,@"str2",str2,@"str2",str3,@"str3", nil];
NSLog(@"dic.count=%d",dic.count); //0

//取值
NSString *lastName = [personData objectForKey:@"lastName"];

//---->字面值取值

NSString *lastName = personData[@"lastName"];

//可变的话可以这样设置新值
[mutableDictionary setObject:@"Galloway" forKey:@"lastName"];

//---->字面值设置值

mutableDictionary[@"lastName"] = @"Galloway";
```

## 效果图

(无)

## 备注

访问的时候数组采用下标方式，字典直接通过 `Key` 来取值，需要注意的就是数组和字典中不能插入 `nil`，否则会抛出异常。

字面量语法可以读写 `mutable` 类型的对象，但通过字面量语法创建的对象是非 `mutable` 的。如果需要将非 `mutable` 类的字面量对象变更为 `mutable` 类型，可以使用 `mutableCopy` 方法。

```
NSMutableArray *mutable = [@[@1, @2, @3, @4, @5] mutableCopy];
```

自定义的字面量语法对象的子类不支持字面量语法。

以下为 Github 上的常用 Swift 字面量转换例子

```
// Literally.swift
//
// Copyright (c) 2014 Mattt Thompson (http://mattt.me)
//
// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"),
// to deal
// in the Software without restriction, including without limitation the rights
// to use, copy, modify, merge, publish, distribute, sublicense,
// and/or sell
// copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
```

```

// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
D, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
ANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN
// THE SOFTWARE.

// MARK: - Foundation -

import Foundation

// MARK: NSCharacterSet

extension NSCharacterSet: ArrayLiteralConvertible {
    public class func convertFromArrayLiteral(characters: Character...) -> Self {
        return self(charactersInString: join("", characters.map({ String($0)})))
    }
}

extension NSCharacterSet: StringLiteralConvertible {
    typealias ExtendedGraphemeClusterLiteralType = StringLiteralType

    public class func convertFromExtendedGraphemeClusterLiteral(
value: StringLiteralType) -> Self {
        return self(charactersInString: value)
    }

    public class func convertFromStringLiteral(value: StringLiteralType) -> Self {
        return self(charactersInString: value)
    }
}

```

```

    }

}

// MARK: NSExpression

extension NSExpression: StringLiteralConvertible {
    typealias ExtendedGraphemeClusterLiteralType = StringLiteral
    Type

    public class func convertFromExtendedGraphemeClusterLiteral(
        value: StringLiteralType) -> Self {
        return self(format: value, argumentArray: [])
    }

    public class func convertFromStringLiteral(value: StringLite
        ralType) -> Self {
        return self(format: value, argumentArray: [])
    }
}

// MARK: NSIndexPath

extension NSIndexPath: ArrayLiteralConvertible {
    public class func convertFromArrayLiteral(indexes: Int... ) -> Self {
        return self(indexes: indexes, length: indexes.count)
    }
}

// MARK: NSIndexSet

extension NSIndexSet: ArrayLiteralConvertible {
    public class func convertFromArrayLiteral(indexes: Int... ) -> Self {
        var mutableIndexSet = NSMutableIndexSet()
        for index in indexes {
            mutableIndexSet.addIndex(index)
        }

        return self(indexSet: mutableIndexSet)
    }
}

```

```
    }

}

// MARK: NSNull

extension NSNull: NilLiteralConvertible {
    public class func convertFromNilLiteral() -> Self {
        return self()
    }
}

// MARK: NSOrderedSet

extension NSOrderedSet: ArrayLiteralConvertible {
    public class func convertFromArrayLiteral(elements: AnyObject ...) -> Self {
        return self(array: elements)
    }
}

// MARK: NSPredicate

extension NSPredicate: StringLiteralConvertible {
    typealias ExtendedGraphemeClusterLiteralType = StringLiteralType

    public class func convertFromExtendedGraphemeClusterLiteral(value: StringLiteralType) -> Self {
        return self(format: value, argumentArray: [])
    }

    public class func convertFromStringLiteral(value: StringLiteralType) -> Self {
        return self(format: value, argumentArray: [])
    }
}

// MARK: NSRegularExpression

extension NSRegularExpression: StringLiteralConvertible {
```

```

typealias ExtendedGraphemeClusterLiteralType = StringLiteral
Type

    public class func convertFromExtendedGraphemeClusterLiteral(
value: StringLiteralType) -> Self {
        return self(pattern: value, options: nil, error: nil)
    }

    public class func convertFromStringLiteral(value: StringLite
ralType) -> Self {
        return self(pattern: value, options: nil, error: nil)
    }
}

// MARK: NSScanner

extension NSScanner: StringLiteralConvertible {
    typealias ExtendedGraphemeClusterLiteralType = StringLiteral
Type

    public class func convertFromExtendedGraphemeClusterLiteral(
value: StringLiteralType) -> Self {
        return self(string: value)
    }

    public class func convertFromStringLiteral(value: StringLite
ralType) -> Self {
        return self(string: value)
    }
}

// MARK: NSSet

extension NSSet: ArrayLiteralConvertible {
    public class func convertFromArrayLiteral(elements: AnyObject
...) -> Self {
        return self(array: elements)
    }
}

```

```
// MARK: NSTimeZone

extension NSTimeZone: StringLiteralConvertible {
    typealias ExtendedGraphemeClusterLiteralType = StringLiteral
    Type

    public class func convertFromExtendedGraphemeClusterLiteral(
        value: StringLiteralType) -> Self {
        return self(name: value)
    }

    public class func convertFromStringLiteral(value: StringLite
    ralType) -> Self {
        return self(name: value)
    }
}

// MARK: NSURL

extension NSURL: StringLiteralConvertible {
    typealias ExtendedGraphemeClusterLiteralType = StringLiteral
    Type

    public class func convertFromExtendedGraphemeClusterLiteral(
        value: StringLiteralType) -> Self {
        return self(string: value)
    }

    public class func convertFromStringLiteral(value: StringLite
    ralType) -> Self {
        return self(string: value)
    }
}

// MARK: - UIKit -

#if os(iOS)

import UIKit
```

```

// MARK: UIColor

extension UIColor: IntegerLiteralConvertible {
    public class func convertFromIntegerLiteral(value: IntegerLiteralType) -> Self {
        let red = CGFloat((value & 0xFF0000) >> 16) / 255.0
        let green = CGFloat((value & 0x00FF00) >> 8) / 255.0
        let blue = CGFloat(value & 0x0000FF) / 255.0
        let alpha = CGFloat(1.0)

        return self(red: red, green: green, blue: blue, alpha: alpha)
    }
}

// MARK: UIImage

extension UIImage: StringLiteralConvertible {
    typealias ExtendedGraphemeClusterLiteralType = StringLiteralType

    public class func convertFromExtendedGraphemeClusterLiteral(
        value: StringLiteralType) -> Self {
        return self(named: value)
    }

    public class func convertFromStringLiteral(value: StringLiteralType) -> Self {
        return self(named: value)
    }
}

#endif

// MARK: - AppKit -

#if os(OSX)

import Cocoa

```

```
extension NSColor: IntegerLiteralConvertible {
    public class func convertFromIntegerLiteral(value: IntegerLiteralType) -> Self {
        let red = CGFloat((value & 0xFF0000) >> 16) / 255.0
        let green = CGFloat((value & 0x00FF00) >> 8) / 255.0
        let blue = CGFloat(value & 0x0000FF) / 255.0
        let alpha = CGFloat(1.0)

        return self(red: red, green: green, blue: blue, alpha: alpha)
    }
}

#endif
```

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-26 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

语法 - 开发常用的宏定义

## 关键字

宏定义\工具类

## 需求场景

1. 简化代码，提高统一，避免出错

## 参考链接

(无)

## 详细内容

### 1. Objective-C 版本

```
#define NavigationBar_HEIGHT 44

#define SCREEN_WIDTH ([UIScreen mainScreen].bounds.size.width)
#define SCREEN_HEIGHT ([UIScreen mainScreen].bounds.size.height)
#define SAFE_RELEASE(x) [x release];x=nil
#define IOS_VERSION [[[UIDevice currentDevice] systemVersion] floatValue]
#define CurrentSystemVersion ([[UIDevice currentDevice] systemVersion])
```

```

#define CurrentLanguage ([[NSLocale preferredLanguages] objectAtIndex:0])

#define BACKGROUND_COLOR [UIColor colorWithRed:242.0/255.0 green:236.0/255.0 blue:231.0/255.0 alpha:1.0]

//use dlog to print while in debug model
#ifndef DEBUG
# define DLog(fmt, ...) NSLog(@"%@", Line %d) __PRETTY_FUNCTION__, __LINE__, ##__VA_ARGS__);
#else
# define DLog(...)
#endif

#define isRetina ([UIScreen instancesRespondToSelector:@selector(currentMode)] ? CGSizeEqualToSize(CGSizeMake(640, 960), [[UIScreen mainScreen] currentMode].size) : NO)
#define iPhone5 ([UIScreen instancesRespondToSelector:@selector(currentMode)] ? CGSizeEqualToSize(CGSizeMake(640, 1136), [[UIScreen mainScreen] currentMode].size) : NO)
#define isPad (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPad)

#if TARGET_OS_IPHONE
//iPhone Device
#endif

#if TARGET_IPHONE_SIMULATOR
//iPhone Simulator
#endif

//ARC
#if __has_feature(objc_arc)
    //compiling with ARC
#else

```

```

// compiling without ARC
#endif

//G-C-D
#define BACK(block) dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), block)
#define MAIN(block) dispatch_async(dispatch_get_main_queue(), block)

#define USER_DEFAULT [NSUserDefaults standardUserDefaults]
#define ImageNamed(_pointer) [UIImage imageNamed:[UIUtil imageName:_pointer]]

#pragma mark - common functions
#define RELEASE_SAFELY(__POINTER) { [__POINTER release]; __POINTER = nil; }

#pragma mark - degrees/radian functions
#define degreesToRadian(x) (M_PI * (x) / 180.0)
#define radianToDegrees(radian) (radian*180.0)/(M_PI)

#pragma mark - color functions
#define RGBCOLOR(r,g,b) [UIColor colorWithRed:(r)/255.0f green:(g)/255.0f blue:(b)/255.0f alpha:1]
#define RGBACOLOR(r,g,b,a) [UIColor colorWithRed:(r)/255.0f green:(g)/255.0f blue:(b)/255.0f alpha:(a)]
#define ITTDEBUG
#define ITTLOGLEVEL_INFO      10
#define ITTLOGLEVEL_WARNING   3
#define ITTLOGLEVEL_ERROR     1

#ifndef ITTMAXLOGLEVEL
#define DEBUG
#define ITTMAXLOGLEVEL ITTLOGLEVEL_INFO

```

```

#else
    #define ITTMAXLOGLEVEL ITTLOGLEVEL_ERROR
#endif

#endif

// The general purpose logger. This ignores logging levels.
#ifdef ITTDEBUG
    #define ITTDPRINT(xx, ...) NSLog(@"%@", @"%s(%d): " xx, __PRETTY_FUNCTION__, __LINE__, ##__VA_ARGS__)
#else
    #define ITTDPRINT(xx, ...) ((void)0)
#endif

// Prints the current method's name.
#define ITTDPRINTMETHODNAME() ITTDPRINT(@"%@", __PRETTY_FUNCTION__)

// Log-level based logging macros.
#if ITTLOGLEVEL_ERROR <= ITTMAXLOGLEVEL
    #define ITTDERROR(xx, ...) ITTDPRINT(xx, ##__VA_ARGS__)
#else
    #define ITTDERROR(xx, ...) ((void)0)
#endif

#if ITTLOGLEVEL_WARNING <= ITTMAXLOGLEVEL
    #define ITTDWARNING(xx, ...) ITTDPRINT(xx, ##__VA_ARGS__)
#else
    #define ITTDWARNING(xx, ...) ((void)0)
#endif

#if ITTLOGLEVEL_INFO <= ITTMAXLOGLEVEL
    #define ITTDINFO(xx, ...) ITTDPRINT(xx, ##__VA_ARGS__)
#else
    #define ITTDINFO(xx, ...) ((void)0)
#endif

#ifdef ITTDEBUG
    #define ITTCONDITIONLOG(condition, xx, ...) { if ((condition)) { \

```

```

    ITTDPRINT(xx,
##__VA_ARGS__); \
} \
} ((void)0)
#else
#define ITTCONDITIONLOG(condition, xx, ...) ((void)0)
#endif

#define ITTAssert(condition, ...)
\
do {
    \
    if (!(condition)) {
        \
        [[NSAssertionHandler currentHandler]
        \
            handleFailureInFunction:[NSString stringWithFormat:@"assertion failed (%@) in %s at line %d", __PRETTY_FUNCTION__, __FILE__, __LINE__]
            \
            file:[NSString stringWithFormat:@"%@", __FILE__]
            \
            lineNumber:__LINE__
            \
            description:__VA_ARGS__];
        \
    }
}
} while(0)

#define _po(o) DLOG(@"%@", (o))
#define _pn(o) DLOG(@"%d", (o))
#define _pf(o) DLOG(@"%f", (o))
#define _ps(o) DLOG(@"CGSize: %@", (o).width, (o).height)
#define _pr(o) DLOG(@"NSRect: {{%.0f, %.0f}, {%.0f, %.0f}}", (o).origin.x, (o).origin.y, (o).size.width, (o).size.height)

#define DOBJ(obj) DLOG(@"%@", #obj, [(obj) description])

```

```
#define MARK      NSLog(@"%@", __PRETTY_FUNCTION__, __LINE__)

#define LOADIMAGE(file,ext) [UIImage imageWithContentsOfFile:[[NSBundle mainBundle]pathForResource:file ofType:ext]]

#define VIEWWITHTAG(_OBJECT, _TAG)      [_OBJECT viewWithTag : _TAG]

// rgb颜色转换 (16进制->10进制)
#define UIColorFromRGB(rgbValue) [UIColor colorWithRed:((float)((rgbValue & 0xFF0000) >> 16))/255.0 green:((float)((rgbValue & 0xFF00) >> 8))/255.0 blue:((float)(rgbValue & 0xFF))/255.0 alpha:1.0]

//RGB颜色
#define RGB(RED, GREEN, BLUE) [UIColor colorWithRed:RED/255.00 green:GREEN/255.00 blue:BLUE/255.00 alpha:1.0]
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

语法 - 弧度 (radians) 和角度转换 (degree)

## 关键字

语法 \ radians \ degree

## 需求场景

- 弧度 (radians) 和角度转换 (degree) 相互转换

## 参考链接

(无)

## 详细内容

- Swift 解决方案

```
CGFloat DegreesToRadians(CGFloat degrees) {return degrees *  
M_PI / 180;};
CGFloat RadiansToDegrees(CGFloat radians) {return radians *  
180 / M_PI;};
```

- Objective-C 解决方案 ``objectivec / **Degrees to Radian** /

**define degreesToRadians( degrees ) ( (**

**degrees ) / 180.0 \* M\_PI )**

/ Radians to Degrees /

**define radiansToDegrees( radians ) ( (  
radians ) \* ( 180.0 / M\_PI ) )**

...

效果图

(无)

备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-22 | Alfred Jiang | -  |

## 方案名称

语法 - 深入理解 GCD

## 关键字

语法 \ GCD

## 需求场景

1. 熟悉 Objective-C GCD

## 参考链接

1. GitHub - nixzhu/dev-blog - GCD 深入理解 : 第一部分
2. GitHub - nixzhu/dev-blog - GCD 深入理解 : 第二部分
3. Ray Wenderlich - Grand Central Dispatch In-Depth: Part 1/2
4. Ray Wenderlich - Grand Central Dispatch In-Depth: Part 2/2

## 详细内容

(见参考链接)

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-21 | Alfred Jiang | -  |

## 方案名称

语法 - 类似 NSError 的引用传值实现

## 关键字

语法 \ NSError \ 引用传值 \ Objective-C 多值返回

## 需求场景

1. 需要实现类似 NSError 的返回时

## 参考链接

(无)

## 详细内容

1. 定义 ``objectiveC
2. -(BOOL)validatePassword:(NSString \*)password

```
failingRules:(out NSArray *__autoreleasing *)rules
```

```
{ NSArray failingRules = [self.rules filteredArrayUsingPredicate:[NSPredicate predicateWithBlock:^BOOL(id rule, NSDictionary bindings) {
```

```
return [rule evaluateWithString:password];
```

```
 }]];
if (rules) {
    *rules = failingRules;
}

return [failingRules count] == 0; } ````
```

### 3. 调用

```
NSArray *failingRules = nil;
if ([self.validator validatePassword:password failingRules:&
failingRules]) {
    //
}
```

## 效果图

(无)

## 备注

(无)

Image\_00001\_20151217\_00001

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-06-16 | Alfred Jiang | -  |

## 方案名称

语法 - 通过 registerDefaults 实现单次记录需求

## 关键字

语法 \ NSUserDefaults \ registerDefaults \ 单次记录

## 需求场景

1. 需要实现单次记录的需求场景，避免重复赋值与重复记录

## 参考链接

1. [CSDN - iOS: UserDefaults issues](#)

## 详细内容

registerDefaults 方法调用时会先检查 UserDefaults 中是否已经存在相同的 Key，如果存在则不会把其覆盖。该特性可以很好的用于记录单次运行的需求，当某 Key 已经被赋值后，避免重复赋值和多次赋值。

示例代码：

```
// [NSUserDefaults standardUserDefaults] 不包含 TEST_KEY
NSLog(@"Log 0 : %@", [[NSUserDefaults standardUserDefaults] objectForKey:TEST_KEY]);

// 通过 registerDefaults: 设置 TEST_KEY 值为 @"TEST_VALUE_0"
NSDictionary *defaultValues0 = [NSDictionary dictionaryWithObjectsAndKeys: @"TEST_VALUE_0", TEST_KEY, nil];
[[NSUserDefaults standardUserDefaults] registerDefaults:defaultValues0];

// 此时 TEST_KEY 值为 @"TEST_VALUE_0"
NSLog(@"Log 1 : %@", [[NSUserDefaults standardUserDefaults] objectForKey:TEST_KEY]);

// 通过 setObject:forKey: 变更 TEST_KEY 值为 @"TEST_VALUE_1"
[[NSUserDefaults standardUserDefaults] setObject:@"TEST_VALUE_1" forKey:TEST_KEY];
[[NSUserDefaults standardUserDefaults] synchronize];

// 此时 TEST_KEY 值为 @"TEST_VALUE_1"
NSLog(@"Log 2 : %@", [[NSUserDefaults standardUserDefaults] objectForKey:TEST_KEY]);

// 通过 registerDefaults: 尝试变更 TEST_KEY 值为 @"TEST_VALUE_2"
NSDictionary *defaultValues1 = [NSDictionary dictionaryWithObjectsAndKeys: @"TEST_VALUE_2", TEST_KEY, nil];
[[NSUserDefaults standardUserDefaults] registerDefaults:defaultValues1];

// 此时 TEST_KEY 值仍然为 @"TEST_VALUE_1"
NSLog(@"Log 3 : %@", [[NSUserDefaults standardUserDefaults] objectForKey:TEST_KEY]);
```

输出内容：

```
2016-06-16 19:50:35.708 testregisterdefaults[34268:3677309] Log
0 : (null)
2016-06-16 19:50:35.709 testregisterdefaults[34268:3677309] Log
1 : TEST_VALUE_0
2016-06-16 19:50:35.713 testregisterdefaults[34268:3677309] Log
2 : TEST_VALUE_1
2016-06-16 19:50:35.713 testregisterdefaults[34268:3677309] Log
3 : TEST_VALUE_1
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-26 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

语法 - 随机数生成实现

## 关键字

语法 \ 随机ID \ 随机字符串 \ 随机数

## 需求场景

1. 测试随机结果的场景需求

## 参考链接

1. [CSDN - ios 生成不重复随机数](#)

## 详细内容

iOS 有如下三种随机数方法：

```
//1.
srand((unsigned)time(0)); //不加这句每次产生的随机数不变
int i = rand() % 5;

//2.
random(time(0));
int i = random() % 5;

//3.
int i = arc4random() % 5 ;
```

注：`rand()` 和 `random()` 实际并不是一个真正的伪随机数发生器，在使用之前需要先初始化随机种子，否则每次生成的随机数一样。

`arc4random()` 是一个真正的伪随机算法，不需要生成随机种子，因为第一次调用的时候就会自动生成。而且范围是 `rand()` 的两倍。在 iPhone 中，`RAND_MAX` 是 `0xffffffff (2147483647)`，而 `arc4random()` 返回的最大值则是 `0x100000000 (4294967296)`。

精确度比较：`arc4random() > random() > rand()`。

常用方法：`arc4random`

1. 获取一个随机整数范围在：[0,100)包括0，不包括100

```
int x = arc4random() % 100;
```

2. 获取一个随机数范围在：[500,1000) ，包括500，不包括1000

```
int y = (arc4random() % 501) + 500;
```

3. 获取一个随机整数，范围在[from,to) ，包括from，不包括to

```
- (int)getRandomNumber:(int)from to:(int)to
{
    return (int)(from + (arc4random() % (to - from + 1)));
}
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-03 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

通知 - iOS7 的多任务处理——远程通知 (Remote Notifications)

## 关键字

通知 \ 多任务处理 \ 后台获取数据 \ Background Fetch \ Remote Notification

## 需求场景

1. 需要获取服务器的通知时

## 参考链接

1. [iOS 7系列译文：iOS7的多任务处理](#)

## 详细内容

1. 注册推送通知 ``objectivec

```
1. (BOOL)application:(UIApplication )application
didFinishLaunchingWithOptions:(NSDictionary )launchOptions {
[[UIApplication sharedApplication] registerForRemoteNotifications]; return
YES; } ``
```
3. 处理一个推送通知 ``objectivec
4. (NSURLSession )backgroundURLSession { static NSURLSession session =
nil; static dispatch\_once\_t onceToken; dispatch\_once(&onceToken, ^{
NSString identifier = @ "io.objc.backgroundTransferExample";

```
NSURLSessionConfiguration sessionConfig = [NSURLSessionConfiguration
backgroundSessionConfiguration:identifier]; session = [NSURLSession
sessionWithConfiguration:sessionConfig delegate:self delegateQueue:
[NSOperationQueue mainQueue]]; });

return session; }
```

5. (void) application:(UIApplication )application didReceiveRemoteNotification:
(NSDictionary )userInfo fetchCompletionHandler:(void (^)
(UIBackgroundFetchResult))completionHandler { NSLog(@"Received remote
notification with userInfo %@", userInfo); NSNumber contentID =
userInfo[@"content-id"]; NSString downloadURLString = [NSString
stringWithFormat:@"http://yourserver.com/downloads/%d.mp3", [contentID
intValue]]; NSURL downloadURL = [NSURL
URLWithString:downloadURLString]; NSURLRequest request =
[NSURLRequest requestWithURL:downloadURL];
NSURLSessionDownloadTask \*task = [[self background URLSession]
downloadTaskWithRequest:request]; task.taskDescription = [NSString
stringWithFormat:@"Podcast Episode %d", [contentID intValue]]; [task
resume]; completionHandler(UIBackgroundFetchResultNewData); } ````
6. 通过实现NSURLSessionDownloadDelegate的委托方法完成下载任务  
```objectiveC

## Pragma Mark - NSURLSessionDownloadDelegate

7. (void) URLSession:(NSURLSession )session downloadTask:
(NSURLSessionDownloadTask )downloadTask didFinishDownloadingToURL:
(NSURL \*)location { NSLog(@"downloadTask:%@
didFinishDownloadingToURL:%@", downloadTask.taskDescription, location);
// Copy file to your app's storage with NSFileManager // ... // Notify your UI }
8. (void) URLSession:(NSURLSession )session downloadTask:
(NSURLSessionDownloadTask )downloadTask didResumeAtOffset:
(int64\_t)fileOffset expectedTotalBytes:(int64\_t)expectedTotalBytes {

}

- (void) URLSession:(NSURLSession )session downloadTask:(NSURLSessionDownloadTask )downloadTask didWriteData:(int64\_t)bytesWritten totalBytesWritten:(int64\_t)totalBytesWritten totalBytesExpectedToWrite:(int64\_t)totalBytesExpectedToWrite {

}

任务完成下载时，你会得到一个磁盘上该文件的临时URL。你必须把这个文件移动或复制到你的应用程序空间，因为当你从这个委托方法返回时，该文件将从临时存储中删除。

4. 当后台会话任务完成后，如果需要打开程序，则实现如下代码

```
```objective-c
- (void) application:(UIApplication *)application
handleEventsForBackgroundURLSession:(NSString *)identifier completionHandler:(void (^)(()()completionHandler
{
    // You must re-establish a reference to the background session,
    // or NSURLSessionDownloadDelegate and NSURLSessionDelegate
methods will not be called
    // as no delegate is attached to the session. See background
URLSession above.
    NSURLSession *backgroundSession = [self backgroundURLSession];
    NSLog(@"Rejoining session with identifier %@ %@", identifier
, backgroundSession);
    // Store the completion handler to update your UI after proc
essing session events
    [self addCompletionHandler:completionHandler forSession:iden
tifier];
}

- (void)URLSessionDidFinishEventsForBackgroundURLSession:(NSURLSession
* )session
{
    NSLog(@"Background URL session %@ finished events.n", sessio
n);
```

```

    if (session.configuration.identifier) {
        // Call the handler we stored in -application:handleEventsForBackground URLSession:
        [self callCompletionHandlerForSession:session.configuration.identifier];
    }
}

- (void)addCompletionHandler:(CompletionHandlerType)handler forSession:(NSString *)identifier
{
    if ([self.completionHandlerDictionary objectForKey:identifier]) {
        NSLog(@"Error: Got multiple handlers for a single session identifier. This should not happen.");
    }
    [self.completionHandlerDictionary setObject:handler forKey:identifier];
}

- (void)callCompletionHandlerForSession: (NSString *)identifier
{
    CompletionHandlerType handler = [self.completionHandlerDictionary objectForKey: identifier];
    if (handler) {
        [self.completionHandlerDictionary removeObjectForKey: identifier];
        NSLog(@"Calling completion handler for session %@", identifier);
        handler();
    }
}

```

不同于以往的委托回调，该应用程序委托会被调用两次，因为您的会话和任务委托可能会收到一系列消息。应用程序委托的

*handleEventsForBackground URLSession* 方法，在这些 *NSURLSession* 委托的消息发送前被调用，然后，

*URLSessionDidFinishEventsForBackground URLSession* 被调用。在前面的方法中，储存了一个后台完成处理代码（completionHandler），并在后面的方法中调用该代码更新界面。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-23 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

通知 - 本地推送通知 (Local Notification) 的测试与实现

## 关键字

通知 \ 推送 \ UILocalNotification \ 本地推送 \ 本地通知 \ 闹钟

## 需求场景

1. 需要实现类似闹铃功能的本地推送通知需求

## 参考链接

1. [GitHub - JRNLocalNotificationCenter](#)
2. [CSDN - iOS本地推送与取消本地通知](#)
3. [CSDN - 闹钟app小结\(ios\)](#)
4. [《本地和推送通知编程指南》](#)

## 详细内容

### 1. JRNLocalNotificationCenter

1. JRNLocalNotificationCenter.h ```objectivec // // JRNLocalNotificationCenter.h  
// DemoApp // // Created by jarinosuke on 7/27/13. // Copyright (c) 2013  
jarinosuke. All rights reserved. //

## import

# import

```
extern NSString const JRNLocalNotificationKeyName; extern NSString  
const JRNAplicationDidReceiveLocalNotification;
```

```
typedef void (^JRNLocalNotificationHandler)(NSString key, NSDictionary  
userInfo);
```

```
@interface JRNLocalNotificationCenter : NSObject @property (nonatomic, copy)  
JRNLocalNotificationHandler localNotificationHandler;
```

- (instancetype)defaultCenter;
- (NSArray \*)localNotifications;

//Handling

- (void)didReceiveLocalNotificationUserInfo:(NSDictionary \*)userInfo;

//Cancel

- (void)cancelAllLocalNotifications;
- (void)cancelLocalNotification:(UILocalNotification \*)localNotification;
- (void)cancelLocalNotificationForKey:(NSString \*)key;

//Post on now

- (UILocalNotification )postNotificationOnNowForKey:(NSString )key

```
    alertBody:(NSString *)a  
lertBody;
```

- (UILocalNotification )postNotificationOnNowForKey:(NSString )key

```
    alertBody:(NSString *)a  
lertBody  
    userInfo:(NSDictionary  
*)userInfo;
```

- (UILocalNotification )postNotificationOnNowForKey:(NSString )key

```
    alertBody:(NSString *)a  
lertBody  
    alertAction:(NSString *)a  
lertAction  
    soundName:(NSString *)s  
oundName  
    launchImage:(NSString *)l  
aunchImage  
    userInfo:(NSDictionary  
*)userInfo  
    badgeCount:(NSUInteger)b  
adgeCount  
    repeatInterval:(NSCalendarUn  
it)repeatInterval;
```

//Post on specified date

- (UILocalNotification )*postNotificationOn:(NSDate )fireDate*

```
        forKey:(NSString *)key  
alertBody:(NSString *)alertBody;
```

- (UILocalNotification )*postNotificationOn:(NSDate )fireDate*

```
        forKey:(NSString *)key  
alertBody:(NSString *)alertBody  
userInfo:(NSDictionary *)userIn  
fo;
```

- (UILocalNotification )*postNotificationOn:(NSDate )fireDate*

```
        forKey:(NSString *)key  
alertBody:(NSString *)alertBody  
userInfo:(NSDictionary *)userIn  
fo  
badgeCount:(NSInteger)badgeCount;
```

- (UILocalNotification )*postNotificationOn:(NSDate )fireDate*

```

        forKey:(NSString *)key
        alertBody:(NSString *)alertBody
        alertAction:(NSString *)alertActio
n
        soundName:(NSString *)soundName
        launchImage:(NSString *)launchImag
e
        userInfo:(NSDictionary *)userIn
fo
        badgeCount:(NSUInteger)badgeCount
repeatInterval:(NSCalendarUnit)repeat
Interval;

```

@end

```

2. JRNLocalNotificationCenter.m
```
objective
//
//  JRNLocalNotificationCenter.m
//  DemoApp
//
//  Created by jarinosuke on 7/27/13.
//  Copyright (c) 2013 jarinosuke. All rights reserved.
//


#import "JRNLocalNotificationCenter.h"

NSString *const JRNLocalNotificationKeyName = @"JRN_KEY"
;
NSString *const JRNApplicationDidReceiveLocalNotification = @"JR
NApplicationDidReceiveLocalNotification";

@interface JRNLocalNotificationCenter()
@property (nonatomic) NSMutableDictionary *localPushDictionary;
@property (nonatomic) BOOL checkRemoteNotificationAvailability;
@end

```

```
static JRNLocalNotificationCenter *defaultCenter;

@implementation JRNLocalNotificationCenter

+ (instancetype)defaultCenter
{
    static dispatch_once_t onceToken;
    dispatch_once(&onceToken, ^{
        defaultCenter = [JRNLocalNotificationCenter new];
        defaultCenter.localPushDictionary = [NSMutableDictionary
new];
        [defaultCenter loadScheduledLocalPushNotificationsFromAp
plication];
        defaultCenter.checkRemoteNotificationAvailability = NO;
        defaultCenter.localNotificationHandler = nil;
    });
    return defaultCenter;
}

- (void)loadScheduledLocalPushNotificationsFromApplication
{
    NSArray *scheduleLocalPushNotifications = [[UIApplication sh
aredApplication] scheduledLocalNotifications];
    for (UILocalNotification *localNotification in scheduleLocal
PushNotifications) {
        if (localNotification.userInfo[JRNLocalNotificationHandl
ingKeyName]) {
            [self.localPushDictionary setObject:localNotificatio
n forKey:localNotification.userInfo[JRNLocalNotificationHandling
KeyName]];
        }
    }
}

- (NSArray *)localNotifications
{
    return [[NSArray alloc] initWithArray:[self.localPushDiction
ary allValues]];
}
```

```
- (void)didReceiveLocalNotificationUserInfo:(NSDictionary *)userInfo
{
    NSString *key = userInfo[JRNLocalNotificationHandlingKeyName];
    if (!key) {
        return;
    }
    [self.localPushDictionary removeObjectForKey:key];

    [[NSNotificationCenter defaultCenter] postNotificationName:JRNApplicationDidReceiveLocalNotification
                                                object:nil
                                                userInfo:userInfo];

    if (self.localNotificationHandler) {
        self.localNotificationHandler(userInfo[JRNLocalNotificationHandlingKeyName], userInfo);
    }
}

- (void)cancelAllLocalNotifications
{
    [[UIApplication sharedApplication] cancelAllLocalNotifications];
    [self.localPushDictionary removeAllObjects];
}

- (void)cancelLocalNotification:(UILocalNotification *)localNotification
{
    if (!localNotification) {
        return;
    }

    [[UIApplication sharedApplication] cancelLocalNotification:localNotification];
}
```

```
    if (localNotification.userInfo[JRNLocalNotificationHandlingKeyName]) {
        [self.localPushDictionary removeObjectForKey:localNotification.userInfo[JRNLocalNotificationHandlingKeyName]];
    }
}

- (void)cancelLocalNotificationForKey:(NSString *)key
{
    if (!self.localPushDictionary[key]) {
        return;
    }

    UILocalNotification *localNotification = self.localPushDictionary[key];
    [[UIApplication sharedApplication] cancelLocalNotification:localNotification];
    [self.localPushDictionary removeObjectForKey:key];
}

#pragma mark -
#pragma mark - Post on now

- (UILocalNotification *)postNotificationOnNowForKey:(NSString *)key
                                            alertBody:(NSString *
)alertBody
{
    return [self postNotificationOnNow:YES
                                fireDate:nil
                                forKey:key
                               alertBody:alertBody
                               alertAction:nil
                               soundName:nil
                               launchImage:nil
                               userInfo:nil
                               badgeCount:0
                               repeatInterval:0];
}
```

```
- (UILocalNotification *)postNotificationOnNowForKey:(NSString *)
)key
    alertBody:(NSString *)alertBody
    userInfo:(NSDictionary *)userInfo
{
    return [self postNotificationOnNow:YES
        fireDate:nil
        forKey:key
        alertBody:alertBody
        alertAction:nil
        soundName:nil
        launchImage:nil
        userInfo:userInfo
        badgeCount:0
        repeatInterval:0];
}

- (UILocalNotification *)postNotificationOnNowForKey:(NSString *
)key
    alertBody:(NSString *)alertBody
    userInfo:(NSDictionary *)userInfo
    badgeCount:(NSInteger)badgeCount
{
    return [self postNotificationOnNow:YES
        fireDate:nil
        forKey:key
        alertBody:alertBody
        alertAction:nil
        soundName:nil
        launchImage:nil
        userInfo:userInfo
        badgeCount:badgeCount
        repeatInterval:0];
}

- (UILocalNotification *)postNotificationOnNowForKey:(NSString *
)key
    alertBody:(NSString *)alertBody
    alertAction:(NSString *)alertAction
    soundName:(NSString *)soundName
```

```
        launchImage:(NSString *)launchImage
        userInfo:(NSDictionary *)userInfo
        badgeCount:(NSUInteger)badgeCount
        repeatInterval:(NSCalendarUnit)repeatInterv
al
{
    return [self postNotificationOnNow:YES
                               fireDate:nil
                               forKey:key
                               alertBody:alertBody
                               alertAction:alertAction
                               soundName:soundName
                               launchImage:launchImage
                               userInfo:userInfo
                               badgeCount:badgeCount
                               repeatInterval:repeatInterval];
}

#pragma mark -
#pragma mark - Post on specified date

- (UILocalNotification *)postNotificationOn:(NSDate *)fireDate
    forKey:(NSString *)key
    alertBody:(NSString *)alertBody
{
    return [self postNotificationOnNow:NO
                               fireDate:fireDate
                               forKey:key
                               alertBody:alertBody
                               alertAction:nil
                               soundName:nil
                               launchImage:nil
                               userInfo:nil
                               badgeCount:0
                               repeatInterval:0];
}

- (UILocalNotification *)postNotificationOn:(NSDate *)fireDate
    forKey:(NSString *)key
    alertBody:(NSString *)alertBody
```

```
    userInfo:(NSDictionary *)userInfo
{
    return [self postNotificationOnNow:NO
                                    fireDate:fireDate
                                    forKey:key
                                    alertBody:alertBody
                                    alertAction:nil
                                    soundName:nil
                                    launchImage:nil
                                    userInfo:userInfo
                                    badgeCount:0
                                    repeatInterval:0];
}

- (UILocalNotification *)postNotificationOn:(NSDate *)fireDate
    forKey:(NSString *)key
    alertBody:(NSString *)alertBody
    userInfo:(NSDictionary *)userInfo
    badgeCount:(NSInteger)badgeCount
{
    return [self postNotificationOnNow:NO
                                    fireDate:fireDate
                                    forKey:key
                                    alertBody:alertBody
                                    alertAction:nil
                                    soundName:nil
                                    launchImage:nil
                                    userInfo:userInfo
                                    badgeCount:badgeCount
                                    repeatInterval:0];
}

- (UILocalNotification *)postNotificationOn:(NSDate *)fireDate
    forKey:(NSString *)key
    alertBody:(NSString *)alertBody
    alertAction:(NSString *)alertAction
    soundName:(NSString *)soundName
    launchImage:(NSString *)launchImage
    userInfo:(NSDictionary *)userInfo
    badgeCount:(NSUInteger)badgeCount
```

```

repeatInterval:(NSCalendarUnit)repeatInterval
{
    return [self postNotificationOnNow:NO
                                fireDate:fireDate
                                forKey:key
                                alertBody:alertBody
                                alertAction:alertAction
                                soundName:soundName
                                launchImage:launchImage
                                userInfo:userInfo
                                badgeCount:badgeCount
                                repeatInterval:repeatInterval];
}

- (UILocalNotification *)postNotificationOnNow:(BOOL)presentNow
                                         fireDate:(NSDate *)fireDate
                                         forKey:(NSString *)key
                                         alertBody:(NSString *)alertBody
                                         alertAction:(NSString *)alertAction
                                         soundName:(NSString *)soundName
                                         launchImage:(NSString *)launchImage
                                         userInfo:(NSDictionary *)userInfo
                                         badgeCount:(NSUInteger)badgeCount
                                         repeatInterval:(NSCalendarUnit)repeatInterval;
{
    if (self.localPushDictionary[key]) {
        //same key already exists
        return self.localPushDictionary[key];
    }

    UILocalNotification *localNotification = [[UILocalNotification alloc] init];
    if (!localNotification) {
        return nil;
    }

    UIRemoteNotificationType notificationType = [[UIApplication sharedApplication] enabledRemoteNotificationTypes];
    if (self.checkRemoteNotificationAvailability && notificationType == UIRemoteNotificationTypeNone) {
}

```

```
        return nil;
    }

BOOL needsNotify = NO;

//Alert
if (self.checkRemoteNotificationAvailability && (notificationType & UIRemoteNotificationTypeAlert) != UIRemoteNotificationTypeAlert) {
    needsNotify = NO;
} else {
    needsNotify = YES;
}
//add key name for handling it.
NSMutableDictionary *userInfoAddingHandlingKey = [NSMutableDictionary dictionaryWithDictionary:userInfo];
[userInfoAddingHandlingKey setObject:key forKey:JRNLocalNotificationHandlingKeyName];
localNotification.userInfo          = userInfoAddingHandlingKey;
localNotification.alertBody        = alertBody;
localNotification.alertAction      = alertAction;
localNotification.alertLaunchImage = launchImage;
localNotification.repeatInterval   = repeatInterval;

//Sound
if (self.checkRemoteNotificationAvailability && (notificationType & UIRemoteNotificationTypeSound) != UIRemoteNotificationTypeSound) {
    needsNotify = NO;
} else {
    needsNotify = YES;
}
if (soundName) {
    localNotification.soundName = soundName;
} else {
    localNotification.soundName = UILocalNotificationDefaultSoundName;
}
```

```
//Badge
if (self.checkRemoteNotificationAvailability && (notificationType & UIRemoteNotificationTypeBadge) != UIRemoteNotificationTypeBadge) {
} else {
    localNotification.applicationIconBadgeNumber = badgeCount;
}

if (needsNotify) {
    if (presentNow && !fireDate) {
        [[UIApplication sharedApplication] presentLocalNotificationNow:localNotification];
    } else {
        localNotification.fireDate = fireDate;
        localNotification.timeZone = [NSTimeZone defaultTimeZone];
        [[UIApplication sharedApplication] scheduleLocalNotification:localNotification];
    }
    [self.localPushDictionary setObject:localNotification forKey:key];
    return localNotification;
} else {
    return nil;
}
}

@end
```

## 2. 使用举例 : REXLocalNotificationManager

```
//
//  REXLocalNotificationManager.swift
//  REX
//
//  Created by Alfred Jiang on 4/23/15.
//  Copyright (c) 2015 REX. All rights reserved.
//
```

```

import UIKit

class REXLocalNotificationManager: NSObject {

    var keysForAuctions : NSMutableArray = []

    class var sharedInstance : REXLocalNotificationManager {
        struct Static {
            static var onceToken : dispatch_once_t = 0
            static var instance : REXLocalNotificationManager? = nil
        }
        dispatch_once(&Static.onceToken) {
            Static.instance = REXLocalNotificationManager()
        }
        return Static.instance!
    }

    func cancelAllAuctionsLocalNotification()
    {
        for key in keysForAuctions
        {
            JRNLocalNotificationCenter.defaultCenter().cancelLocalNotificationForKey(key as String)
        }
    }

    func addNotificationsForList(auctions : NSArray)
    {
        self.cancelAllAuctionsLocalNotification()

        //        let keyEnded : String = "Test_Auction_End_Time"
        //        self.addPost(keyEnded, aTime: NSDate(timeIntervalSinceNow:30.0), aBody: "Test Auction ended")

        for obj in auctions
        {
            var aAuction = obj as Auction

            if aAuction.aAuctionStatus == "0" //进行中

```

```

    {
        let keyEnded : String = "Auction_End_Time_\$(aAuction.aId)"
            self.addPost(keyEnded, aTime: aAuction.endDateValue(0), aBody: "Auction ended")

            let keyEnded_1_Hour : String = "Auction_End_1_Hour_Time_\$(aAuction.aId)"
                self.addPost(keyEnded_1_Hour, aTime: aAuction.endDateValue(60 * 60), aBody: "\$(aAuction.aName) will end in 1 hour")

            let keyEnded_3_Hour : String = "Auction_End_3_Hour_Time_\$(aAuction.aId)"
                self.addPost(keyEnded_3_Hour, aTime: aAuction.endDateValue(3 * 60 * 60), aBody: "\$(aAuction.aName) will end in 3 hours")

            let keyEnded_24_Hour : String = "Auction_End_24_Hour_Time_\$(aAuction.aId)"
                self.addPost(keyEnded_24_Hour, aTime: aAuction.endDateValue(24 * 60 * 60), aBody: "\$(aAuction.aName) will end in 24 hours")

    }

    else if aAuction.aAuctionStatus == "1" //未开始
    {
        let keyEnded : String = "Auction_Start_Time_\$(aAuction.aId)"
            self.addPost(keyEnded, aTime: aAuction.startDateValue(0), aBody: "Auction start")

            let keyEnded_1_Hour : String = "Auction_Start_1_Hour_Time_\$(aAuction.aId)"
                self.addPost(keyEnded_1_Hour, aTime: aAuction.startDateValue(60 * 60), aBody: "\$(aAuction.aName) will start 1 hour later")

            let keyEnded_3_Hour : String = "Auction_Start_3_Hour_Time_\$(aAuction.aId)"

```

```
        self.addPost(keyEnded_3_Hour, aTime: aAuction.startDateValue(3 * 60 * 60), aBody: "\(aAuction.aName) will start 3 hours later")

        let keyEnded_24_Hour : String = "Auction_Start_24_Hour_Time_\(" + aAuction.aId + ")"
        self.addPost(keyEnded_24_Hour, aTime: aAuction.startDateValue(24 * 60 * 60), aBody: "\(aAuction.aName) will start 24 hours later")
    }
}

func addPost(aKey : NSString, aTime : NSDate, aBody : NSString)
{
    let dateNow : NSDate = NSDate()
    let overTime : Bool = aTime.timeIntervalSince1970 - dateNow.timeIntervalSince1970 > 0

    if !overTime
    {
        return
    }

    JRNLocalNotificationCenter.defaultCenter().postNotificationOn(aTime, forKey: aKey, alertBody: aBody)
    self.keysForAuctions.addObject(aKey)
}
}
```

### 3. 接受通知

```

func application(application: UIApplication, didReceiveLocalNotification notification: UILocalNotification) {
    println("Local Notification userInfo==\(notification.userInfo)")
    let body : String = notification.alertBody! as String
    var alertView : UIAlertView = UIAlertView(title: "", message: body, delegate: nil, cancelButtonTitle: "OK")
    alertView.show()
}

```

#### 4. 闹钟实现注意事项

使用 *LocalNotification* 实现闹钟时需要注意以下事项

1. 本地的 *apns* 最多只能设置 64 个，如果超过了的话，以最新设置的 64 个 *apns* 为准，最早设置的会被忽略，这个当初没有仔细看文档，导致了不该出现的 *bug*，后文会讲到
2. 当在后台收到 *apns* 时，发出的 *apns* 声音不能超过 30s，如果指定的 *apns* 的 *sound* 文件的实际播放声音超过 30s，则只播放 30s
3. 删掉 *app* 重新下载后，原先设置的本地 *apns* 还在，这个当时也没注意到，吃了大亏，后来一查才知，本地的 *apns* 是系统进程，不会随着你的 *app* 删掉而消失，不过如果你的 *app* 被删除，原先存在的 *apns* 是不会被触发的！
4. 闹钟的功能是以叫醒人为主，但是很遗憾，每个 *UILocalization* 的声音最多只能响一次，*app* 推出后，不少用户也抱怨此功能不人性化，现在我们的做法是将 64 个本地的 *apns* 全部分配给最近的那个闹钟，比如我设置在 13:00 响铃，声音的实际播放时间为 30s，这样每隔 30s 推送一次本地的 *apns*，这样 64 次就足以将用户叫醒（当然如果在此期间用户没将手机带在身边，那就悲剧了，下次就不会再响了，不过此概率极小），如果用户点击 *apns* 进入 *app*，那就 *cancel* 全部的 *UILocalization*，然后再重新设置最新的闹钟，将 64 个 *apns* 全部赋予此闹钟
5. 计算最近的闹钟有一个问题，如何判定这个最近？比如有的闹钟是今天，有的闹钟是昨天（但此闹钟设置为每天响铃），我的做法是，将所有重复闹钟响铃时间的年月日全部换成今天的年月日（时分秒不变），然后再与 *today[NSDate date]* 进行比较，如果比 *today* 早，则将响铃时间设置为之后最近的会响铃的那一天，这样设置之后，就可以获取最近的闹钟时间了

6. 何时重新设置这个闹钟时间，通常的做法是在收到 *UILocalization* (即 *application:didReceiveLocalNotification:*)时重新设置，但是这有一个问题，此方法只有当用户点了 *apns* 的提示时（或在锁屏状态下滑动进入 *app*）时才能触发，也就是说，如果在后台收到了 *apns*，但用户是通过点击 *app* 的 *icon* 进入 *app* 的话，此方法是不会触发的！这也是苹果的一个 *bug* ,所以我的做法是在 *application:didReceiveLocalNotification:* 此方法里触发，也在 *applicationDidBecomeActive* 方法里触发！这样就可以解决此问题了

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-04-22 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

通知 - 远程推送通知 (Remote Notification) 的测试与实现

## 关键字

推送 \ Remote Notification \ 远程推送 \ 远程通知

## 需求场景

1. 需要对 APP 的推送功能进行开发与测试时

## 参考链接

1. 博客园 - iOS推送小结--swift语言
2. CocoaChina - (APNS) IOS远程推送测试工具SmartPush for Mac
3. 使用pushmebaby测试app的远程推送功能
4. 博客园 - Java实现IOS推送 (Javapns2.2)
5. 博客园 - 苹果推送通知服务教程 Apple Push Notification Services Tutorial
6. iPhone推送通知功能分析总结
7. 《本地和推送通知编程指南》

## 详细内容

### 1. iOS 端推送的注册与实现

1. 注册相关推送证书(参考链接)

2. 在 *AppDelegate* 中实现以下步骤

```
swift //1. 在
didFinishLaunchingWithOptions 函数中调用下面函数注册推送 func
registerPush() { if ((UIDevice.currentDevice().systemVersion as
NSString).doubleValue >= 8.0) {
```

```
    UIApplication.sharedApplication().registerForRemoteNotifica
    tions()
    UIApplication.sharedApplication().registerUserNotificationS
    ettings(UIUserNotificationSettings(forTypes:UIUserNotificati
    onType.Badge|UIUserNotificationType.Sound|UIUserNotificati
    onType.Alert, categories: nil))
```

```
} else {
```

```
    UIApplication.sharedApplication().registerForRemoteNotifica
    tionTypes(UIRemoteNotificationType.Badge|UIRemoteNotificatio
    nType.Alert|UIRemoteNotificationType.Sound)
```

```
} }
```

```
//2. 实现下面三个 AppDelegate 函数 func application(application: UIApplication,
didRegisterForRemoteNotificationsWithDeviceToken deviceToken: NSData) { var
token:String =
deviceToken.description.stringByTrimmingCharactersInSet(NSCharacterSet(chara
ctersInString: "<>")) println("token==(token)") //将token发送到服务器 }
```

```
func application(application: UIApplication,
didFailToRegisterForRemoteNotificationsWithError error: NSError) { var
alert:UIAlertView = UIAlertView(title: "", message: error.localizedDescription,
delegate: nil, cancelButtonTitle: "OK") alert.show() //注册失败提示 }
```

```
func application(application: UIApplication, didReceiveRemoteNotification
userInfo: [NSObject : AnyObject]) { println("userInfo==(userInfo)") //成功接收到推
送，对应用内发送通知
NSNotificationCenter.defaultCenter().postNotificationName(DID_RECEIVE_REM
OTE_NOTIFICATION, object: nil, userInfo: userInfo) }
```

### 3. 在需要接收通知的类中实现接收监听

```
```swift
//1. 在 viewDidLoad 函数中开启监听
NSNotificationCenter.defaultCenter().addObserver(self, selector:
    Selector("receiveNotification:"), name: DID_RECEIVE_REMOTE_NOTIFICATION, object: nil)

//2. 在 viewDidLoad 函数中开启监听
func receiveNotification(aNotification : NSNotification)
{
    //收到通知，对应处理操作
}
```

## 1. 脚标清零

```
func applicationDidBecomeActive(application: UIApplication)
{
    application.applicationIconBadgeNumber = 0
}
```

## 2. 远程推送通知的测试与发送

1. 推荐使用 [SmartPush](#) 具体使用步骤可参考[\(APNS\) IOS远程推送测试工具 SmartPush for Mac](#)

## 3. 其他

### 1. 推送格式举例：

```
{"aps" : { "alert" : { "body" : "Bob wants to play poker" }, "badge" : 5, "sound" :"bingbong.aiff"},      "acme1" : "bar", "acme2" : ["bang", "whiz"] }
```

## 2. 注意

- device token 会在重装系统等特殊条件下发生改变，所以需要每次都通过 `didRegisterForRemoteNotificationsWithDeviceToken` 注册获取

- 推送通知的字节长度不能超过256个字节（256B）
- 推送通知不都是可靠的，苹果公司不保证传送每个通知以及通知到达的顺序。
- 当应用未启动时，仅接受并显示 "alert" 相关内容
- 当应用启动时，完整推送消息会被 didReceiveRemoteNotification 函数接受并通过 userInfo 参数获取

### 3. 失败情况

- 苹果公司的反馈服务会报告通知的失败。
- 发布的接口是通过访问：feedback.push.apple.com, port2196。
- 开发的接口是通过访问：feedback.sandbox.push.apple.com, port2196。
- 注：Provider要反复监测苹果的反馈服务，来确保不向已经不存在的设备发送推送通知。

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-26 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

键盘 - 使用 IQKeyboardManager 完美解决IOS开发键盘遮挡

## 关键字

键盘 \ Keyboard \ 弹出键盘 \ 收起键盘

## 需求场景

1. 需要弹出键盘收起键盘的页面避免遮挡问题

## 参考链接

1. [IQKeyboardManager-不用写一行代码就完美解决IOS开发键盘遮挡的类库](#)
2. [GitHub - IQKeyboardManager](#)
3. [GitHub - CocoaPods](#)

## 详细内容

### 1. Cocoapod安装：

你可以使用 [Cocoapods](#) 来安装 [IQKeyboardManager](#) 类库。在 [Podfile](#) 文件中这样写：

```
pod 'IQKeyboardManager'
```

就可以了

## 2. Framework加入:

将 `KeyboardManager.framework`、`IQKeyboardManager.bundle` 等文件加入到项目中即可。详细可以[下载Demo](#)并查看。

注意：需要在项目的设置 `other linker flag` 中加入 `-ObjC`。

建议：目前 [Cocoapods](#) 已经是很成熟的第三方类库管理工具了，推荐使用。

## 3. 常用的属性和方法介绍

- **+sharedManager** : 获取类库的单例变量。我们 also 知道，一个项目中都是使用一个类库的单例的，不然每一个输入框我们怎么好控制呢？所以如果你想自己修改一下界面，那么就要先获取到这个单例的变量，然后在往下面操作。
- **enable** : 这个属性就是说，我们的项目里面使用不适用这个类库所提供的输入框不遮挡技术。如果您再某些页面里面不需要，可以在获取到单例之后，将这个 `enable` 变量设置为 `FALSE`。
- **keyboardDistanceFromTextField** : 这个就是我们的输入框距离我们的键盘的距离了。默认是 `10px`。就是说输入框默认会自动移动到键盘的上面 `10` 个像素以方便用户输入。如果你需要自定义，可以改变这个值。
- **enableAutoToolbar** : `IQKeyboardManager` 提供的键盘上面默认会有“前一个”“后一个”“完成”这样的辅助按钮。如果你不需要，可以将这个 `enableAutoToolbar` 属性设置为 `NO`，这样就不会显示了。
- **toolbarManageBehaviour** : 如果有多个输入框，那么我们在输入的时候可以通过点击在 `Toolbar` 中的“前一个”“后一个”按钮来实现移动到不同的输入框。可是输入框的移动肯定是有一个规律的。这里就提供了两个方式。第一种就是加入的顺序，第二种就是按照 `Tag` 值的大小排序。这个属性可以设置两个参数：`IQAutoToolbarBySubviews` 和 `IQAutoToolbarByTag`。
- **shouldToolbarUsesTextFieldTintColor** : 这个是用来将输入框的 `tintColor` 和 `toolbar` 的 `tintColor` 相互协调的，默认为 `NO`。
- **shouldShowTextFieldPlaceholder** : 如果输入框有 `placeholder` 的话，那么在 `toolbar` 中默认会显示出来。在中间的部分会显示 `UITextField` 的 `placeholder`。如果你不需要，可以设置 `NO`。

- **placeholderFont** : 这个就是 *toolbar* 中显示 *placeholder* 的字体大小了。你可以自定义通过传入一个 *font* 。
- **canAdjustTextView** : 这样说，如果你的输入框有 600px 的高度。那么在点击输入框的时候，键盘弹出来了，输入框会如何显示呢？如果把这个参数打开，那么输入框的高度会刚好的降低，就是说，你可以看到输入框的四个 *board*，操作一下就会一目了然：）

## 效果图

(无)

## 备注

- 键盘 - 弹出与收起改变页面高度

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-02 | Alfred Jiang | -  |
| 2  | 2015-12-22 | Alfred Jiang | -  |

## 方案名称

键盘 - 弹出与收起改变页面高度

## 关键字

键盘 \ Keyboard \ 弹出键盘 \ 收起键盘

## 需求场景

1. 页面弹出键盘需要改变页面高度时

## 参考链接

(无)

## 详细内容

1. Swift 解决方案 ``swift deinit{

```
NSNotificationCenter.defaultCenter().removeObserver(self, name:  
UIKeyboardWillShowNotification, object: nil)  
NSNotificationCenter.defaultCenter().removeObserver(self, name:  
UIKeyboardWillHideNotification, object: nil) }
```

```
func setupNotification() {
```

```
NSNotificationCenter.defaultCenter().addObserver(self, selector:Selector("keyboar  
dDidShow:"),name:UIKeyboardWillShowNotification,object:nil)
```

```
NSNotificationCenter.defaultCenter().addObserver(self, selector:Selector("keyboar dWillHide:"),name:UIKeyboardWillHideNotification,object:nil) }
```

```
//Keyboard did show func keyboardDidShow(sender: NSNotification) { if let keyboardSize = (sender.userInfo?[UIKeyboardFrameEndUserInfoKey] as? NSValue)?.CGRectValue()//此处需要注意一定要是 UIKeyboardFrameEndUserInfoKey { let duration : NSTimeInterval = NSTimeInterval(sender.userInfo?[UIKeyboardAnimationDurationUserInfoKey] as NSNumber)
```

```
        self.tableViewMain.layoutIfNeeded()
        UIView.animateWithDuration(duration, animations: { () -> Void in
            self.tableViewMain.contentInset.bottom = keyboardSize.height
            self.tableViewMain.layoutIfNeeded()
            self.showKeyboard = true
        })
    }
}
```

```
//keyboard will hide func keyboardWillHide(sender: NSNotification){ if let keyboardSize = (sender.userInfo?[UIKeyboardFrameBeginUserInfoKey] as? NSValue)?.CGRectValue() { let duration : NSTimeInterval = NSTimeInterval(sender.userInfo?[UIKeyboardAnimationDurationUserInfoKey] as NSNumber)
```

```
        self.tableViewMain.layoutIfNeeded()
        UIView.animateWithDuration(duration, animations: { () -> Void in
            self.tableViewMain.contentInset.bottom = 0
            self.tableViewMain.layoutIfNeeded()
            self.showKeyboard = false
        })
    }
}
```

```

2. Objective-C 解决方案
```
- (void)dealloc
{
    [[NSNotificationCenter defaultCenter] removeObserver:self name:UIKeyboardWillShowNotification object:nil];
    [[NSNotificationCenter defaultCenter] removeObserver:self name:UIKeyboardWillHideNotification object:nil];
}

- (void)setupNotification
{
    [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(keyboardWillShow:) name:UIKeyboardWillShowNotification object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(keyboardWillHide:) name:UIKeyboardWillHideNotification object:nil];
}

- (void)keyboardWillShow:(NSNotification*)notification {
    _keyboardRect = [[[notification userInfo] objectForKey:_UIKeyboardFrameEndUserInfoKey] CGRectValue];
    _keyboardVisible = YES;

    // Shrink view's inset by the keyboard's height, and scroll
    // to show the text field/view being edited
    [UIView beginAnimations:nil context:NULL];
    [UIView setAnimationCurve:[[[notification userInfo] objectForKey:_UIKeyboardAnimationCurveUserInfoKey] intValue]];
    [UIView setAnimationDuration:[[[notification userInfo] objectForKey:_UIKeyboardAnimationDurationUserInfoKey] floatValue]];

    self.view_center.frame = CGRectMake(0, 0, 320, 250);

    [UIView commitAnimations];
}

- (void)keyboardWillHide:(NSNotification*)notification {
```

```

```
_keyboardRect = CGRectMakeZero;
_keyboardVisible = NO;

// Restore dimensions to prior size
[UIView beginAnimations:nil context:NULL];
[UIView setAnimationCurve:[[[notification userInfo] objectForKey:UIKeyboardAnimationCurveUserInfoKey] intValue]];
[UIView setAnimationDuration:[[[notification userInfo] objectForKey:UIKeyboardAnimationDurationUserInfoKey] floatValue]];

self.view_center.frame = CGRectMake(0, 100, 320, 250);

[UIView commitAnimations];
}
```

## 效果图

(无)

## 备注

- 键盘 - 使用 IQKeyboardManager 完美解决IOS开发键盘遮挡

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-03-20 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

音视频 - 使用 POVoiceHUD 实现语音录制

## 关键字

音视频 \ Voice \ 语音录制 \ POVoiceHUD \ 音频

## 需求场景

1. 需要实现语音录制需求时

## 参考链接

1. [GitHub - POVoiceHUD](#)

## 详细内容

1. 引入以下 **framework**

```
AVFoundation.framework
AudioToolbox.framework
CoreGraphics.framework
QuartzCore.framework
```

2. 在 **ViewDidLoad** 中实例化

```
self.voiceHud = [[POVoiceHUD alloc] initWithParentView:self.view];
self.voiceHud.title = @"Speak Now";

[self.voiceHud setDelegate:self];
[self.view addSubview:self.voiceHud];
```

**3.** 在需要录制的地方调用下面的代码

```
[self.voiceHud startForFilePath:[NSString stringWithFormat:@"%@/Documents/MySound.caf", NSHomeDirectory()]];
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-02-02 | Alfred Jiang | -  |

## 方案名称

音视频 - 使用 TheAmazingAudioEngine 框架实现 iOS 音频开发中的各种音效实现

## 关键字

音视频 \ 音频制作 \ 音效 \ 播放 \ 录音 \ 录制

## 需求场景

1. 实现iOS音频开发中的各种音效的实现

## 参考链接

(见详细内容)

## 详细内容

- 简书 - iOS第三方音频框架TheAmazingAudioEngine使用及音效实现介绍
- GitHub - TheAmazingAudioEngine

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2015-05-05 | Alfred Jiang | -  |
| 2  | 2015-12-23 | Alfred Jiang | -  |

## 方案名称

音视频 - 音频播放、录音、视频播放、拍照、视频录制

## 关键字

音视频 \ 音频播放 \ 录音 \ 视频播放 \ 拍照 \ 视频录制

## 需求场景

1. 需要进行 iOS 音视频录制播放需求的场景

## 参考链接

(见详细内容)

## 详细内容

1. 博客园 - iOS开发系列--音频播放、录音、视频播放、拍照、视频录制

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-23 | Alfred Jiang | -  |

## 方案名称

Python - Mac 下安装 Python 模块

## 关键字

Python \ Mac \ 模块

## 需求场景

1. 需要安装 Python 模块时

## 参考链接

1. 学步园 - Mac 下如何安装 Python 模块

## 详细内容

### 一、使用 **easy\_install** 在线安装：

mac 下默认的 python 环境集成了这个工具可以非常方便的安装各种 python 包。**easy\_install** 是由 PEAK(Python Enterprise Application Kit) 开发的 **setuptools** 包里带的一个命令，所以使用 **easy\_install** 实际上是在调用 **setuptools** 来完成安装模块的工作。

使用方法非常简单，在命令行输入

\$ easy\_install 参数

即可。

`setuptools` 会自动搜索 PyPI 以查找最新版本的模块。如果找到的话，她会自动下载、编译和安装：

```
| $ sudo easy_install easygui
```

二、将模块下载后进行安装：

下载好 `easygui` 后，`cd` 进入解压好的文件夹，

```
| $ python setup.py install
```

即可。

或者使用 `easy_install` 安装也可，`cd` 进入解压好的文件夹，

```
| $ easy_install .
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-14 | Alfred Jiang | -  |

## 方案名称

Python - Python 工具收集

## 关键字

Python \ 脚本 \ 工具 \ 收集

## 需求场景

1. 学习一些有趣的开源脚本和工具

## 参考链接

(见详细内容)

## 详细内容

### 类别一

- 知乎 - 哪些 Python 库让你相见恨晚？
- GitHub - dloss/python-pentest-tools

### 类别二

| 名称                     | 链接                     | 简介                           |
|------------------------|------------------------|------------------------------|
| Erma-Wang/Spider       | <a href="#">GitHub</a> | 放养的小爬虫--京东定向爬虫(AJAX获取价格数据)   |
| objc_dep               | <a href="#">Github</a> | 检查 Objective-C 项目工程中的导入依赖    |
| Python                 | <a href="#">Github</a> | 一些 python 小脚本                |
| sanguo                 | <a href="#">Github</a> | 使用python 和 pygame 编写的三国策略游戏. |
| Similarity Py          | <a href="#">Github</a> | 距离算法                         |
| soimort/you-get        | <a href="#">GitHub</a> | You-Get是一个基于Python 3的视频下载工具  |
| wechat-deleted-friends | <a href="#">Github</a> | 查看被删的微信好友                    |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-04-23 | Alfred Jiang | -  |

## 方案名称

Python - 使用 Anaconda 进行 Python 科学计算

## 关键字

Python \ Anaconda \ 科学计算 \ continuum

## 需求场景

1. 打包安装 Python + NumPy + SciPy + Matplotlib + IPython + IPython Notebook 实现科学计算

## 参考链接

1. [哔哩哔哩 - Python 光速上手](#)
2. [continuum.io](#)

## 详细内容

(见参考链接)

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-23 | Alfred Jiang | -  |

## 方案名称

Python - 使用 virualenv 创建虚拟 Python 环境

## 关键字

Python \ 虚拟机 \ 虚拟环境 \ virualenv

## 需求场景

1. 让同一台电脑运行多个不同版本的 Python 程序

## 参考链接

1. [ITEYE - 使用virualenv创建虚拟python环境](#)

## 详细内容

### 1. 安装 pip

```
$ sudo easy_install pip
```

### 2. 安装 virtualenv

```
$ sudo pip install virtualenv
```

### 3. 初始化虚拟机环境

进入需要初始化 Python 虚拟机环境的目录，执行以下命令

```
$ virtualenv venv
```

如果报出 *virtualenv: command not found* 错误，

可通过 find 命令查找到 virtualenv 安装目录

```
$ find / -name virtualenv
```

得出在 : /usr/local/python2.7.8/bin/virtualenv 目录下

再执行

```
$ /usr/local/python2.7.8/bin/virtualenv venv
```

即可创建虚拟环境 venv

#### 4. 进入虚拟机环境

执行

```
$ cd venv
```

```
$ source bin/activate
```

你会发现在 shell 提示符前面多了个 (venv), 说明已经在虚拟环境中了，再安装对应的工具包，如 flask 等。

#### 5. 其它相关命令：

退出： deactivate

切换到其它环境： workon venv 或 venv2 ...

删除环境： rmvirtualenv venv

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-14 | Alfred Jiang | -  |

## 方案名称

Python - 学习资源收集

## 关键字

专题 \ Python \ 学习资源

## 需求场景

1. Python 学习资源收集参考

## 参考链接

(见详细内容)

## 详细内容

| 名称                 | 链接                                                    | 简介                                                                                                   |
|--------------------|-------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| Python 2.7.11 官方教程 | <a href="#">Python 中文学习大本营 - Python 2.7.11 官方教程</a>   | The Python Tutorial (Python 2.7.11) 的中文翻译版本。Python Tutorial 为初学 Python 必备官方教程，本教程适用于 Python 2.7.X 系列 |
| Python 3.5.1 官方教程  | <a href="#">Python 中文学习大本营 - Python 3.5.1 官方教程</a>    | The Python Tutorial (Python 3.5.1) 的中文翻译版本。Python Tutorial 为初学 Python 必备官方教程，本教程适用于 Python 3.5.x     |
| Python 光速上手        | <a href="#">哔哩哔哩 - Python 光速上手</a>                    | 【散沙】81分钟！Python基础系列教程                                                                                |
| Python 教程(by 廖雪峰)  | <a href="#">廖雪峰的官方网站 - Python 教程</a>                  | Python 基础教程                                                                                          |
| PyTips             | <a href="#">Segmentfault - PyTips</a>                 | <a href="https://git.io/pytips">https://git.io/pytips</a> 每天一条有用的 Python 小提示                         |
| 关于 Python 的面试题     | <a href="#">Github - taizilongxu/interview_python</a> | 关于Python的面试题                                                                                         |
| 《跟老齐学 Python》      | <a href="#">GitHub - qiwsir/StarterLearningPython</a> | 原名叫做《零基础学Python》，后来由于图书出版，更名为《跟老齐学 Python》                                                           |

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-23 | Alfred Jiang | -  |

## 方案名称

Python - 用 Python 和 py2app 写独立的 Mac OS X 应用

## 关键字

Python \ py2app \ Mac OS \ APP

## 需求场景

1. 使用 Python 和 py2app 创建 Mac App

## 参考链接

1. 简书 - [译]用Python和py2app写独立的Mac OS X 应用
2. Chris Hager - Creating standalone Mac OS X applications with Python and py2app

## 详细内容

具体步骤见参考链接，首次操作中遇到以下两个问题

### 1. *Permission denied*

使用 `sudo` 输入管理员密码即可

### 2. *-bash: py2applet: command not found*

执行

```
$ find / -name py2applet
```

找到

`/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/bin/py2applet`

直接通过 `py2applet` 完整路径执行命令即可

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-06-03 | Alfred Jiang | -  |

## 方案名称

Python - 编码规范

## 关键字

Python \ 编码规范

## 需求场景

- 统一团队编码风格，提高工程代码可读性与可维护性

## 参考链接

(见详细内容)

## 详细内容

- PEP 8 -- Style Guide for Python Code
- Hitchhiker's Guide to Python - Code Style

## 效果图

(无)

## 备注

(无)



## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-29 | Alfred Jiang | -  |
| 2  | 2016-01-30 | Alfred Jiang | -  |

## 方案名称

JavaScript - 开发工具介绍

## 关键字

JavaScript \ Mac \ 开发工具

## 需求场景

1. 开始入门学习 JavaScript 相关

## 参考链接

1. [Lynda - JavaScript Essential Training](#)

## 详细内容

### IDE

- [Aptana Studio 3](#)
- [Sublime Text](#)

### 调试工具

- [Firebug](#)

### 验证工具

- [JSLint](#)

## 压缩工具

- [Uglify](#)
- [YUI Compressor](#)
- [Google Closure Compiler](#)

## 在线文档

- [MDN - JavaScript](#)
- [Dev.Opera](#)
- [jQuery](#)
- [Yahoo - JavaScript™ Developer Center](#)

## 其他

- [Google Developers Hosted Libraries](#) : Google 服务器提供的各种 JavaScript 框架和库
- [Modernizr](#) : 检测浏览器对 CSS3 或 HTML5 功能支持情况
- [Can I Use](#) : 一个提供HTML5,CSS3,JS等新技术浏览器兼容一览表的网站

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-30 | Alfred Jiang | -  |

## 方案名称

JavaScript - JavaScript To Avoid

## 关键字

JavaScript \ 编码建议

## 需求场景

- 避免出现在代码中的一些编码习惯

## 参考链接

- [Lynda - JavaScript Essential Training](#)

## 详细内容

因为一些版本原因而应该尽量避免出现的编码实现

- `document.write()`
- `Browser Sniffing` `javascript if (navigator.userAgent.indexOf('Netscape')) ...`

or

```
if (navigator.appName = 'Microsoft Internet Explorer') ...
```

3. eval()

4. Pseudo-Protocols

```html

<p>Inside your HTML, you may find:

<a href="javascript:someFunction()">this</a>

</p>

<p> A preferable way is

<a href="nojavascript.html"

    onclick="someFunction(); return false;">this</a>

</p>

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-05-13 | Alfred Jiang | -  |

## 方案名称

JavaScript - 使用 PDFObject 实现 PDF 文件在线预览

## 关键字

JavaScript \ 在线加载 PDF \ HTML

## 需求场景

1. 需要在线加载 pdf 文档时

## 参考链接

1. [PDFObject](#)
2. [GitHub - PDFObject](#)

## 详细内容

示例：通过 `CN_Blocks_Programming_Topics.html` 加载  
`CN_Blocks_Programming_Topics.pdf`

## 目录结构

```
rootpath
└── ebooks
    └── CN_Blocks_Programming_Topics.html
└── pdfviewer
    └── PDFObject Source Code
└── pdfs
    └── CN_Blocks_Programming_Topics.pdf
```

### CN\_Blocks\_Programming\_Topics.html 代码

```
<!DOCTYPE html>
<html lang="en-us">
<body>
<script src="../pdfviewer/pdfobject.min.js"></script>
<Script Language="JavaScript">
PDFObject.embed("../pdfs/CN_Blocks_Programming_Topics.pdf");
</script>
</body>
</html>
```

## 效果图

(无)

## 备注

(无)

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-01-30 | Alfred Jiang | -  |

## 方案名称

JavaScript - 正则表达式介绍

## 关键字

JavaScript \ 正则表达式

## 需求场景

1. 学习 JavaScript 正则表达式相关

## 参考链接

1. [Lynda - JavaScript Essential Training](#)
2. [正则总结：JavaScript中的正则表达式](#)

## 详细内容

创建 RegExp

```
var myRE = /hello/;

//or

var myRE = new RegExp("hello");

var myString = "Does this sentence have the word hello in it"
if ( myRE.test(myString) ){
    alert("Yes");
}
```

### 符号介绍

```
var myRE = /^hello/;      // ^ at the start
= /hello$/;      // $ at the end
= /hel+o/;      // + once or more
= /hel*o/;      // * zero or more
= /hel?o/;      // ? zero or one
= /hello|goodbey/;      // either|or
= /he..o/;      // . any character
= /\wello/;      // \w alphanumeric or _
= /\bhello/;      // \b word boundary
= /[crnld]ope/;      // [...] range of chars
```

### 效果图

(无)

### 备注

- 正则表达式 - 在 iOS 开发中使用正则表达式

## 变更记录

| 序号 | 录入时间       | 录入人          | 备注 |
|----|------------|--------------|----|
| 1  | 2016-03-14 | Alfred Jiang | -  |

## 方案名称

专题 - JavaScript 学习资源收集

## 关键字

专题 \ JavaScript \ 学习 资源

## 需求场景

1. JavaScript 学习资源收集参考

## 参考链接

(见详细内容)

## 详细内容

| 名称                | 链接                     | 简介                                                         |
|-------------------|------------------------|------------------------------------------------------------|
| Geodesy functions | <a href="#">Github</a> | 地理坐标相关                                                     |
| GitBook Editor    | <a href="#">Github</a> | GitBook 编辑器                                                |
| mama2             | <a href="#">Github</a> | 妈妈计划用于解决在看视频网站时 MacBook 发热严重的问题。使用 video 来替换原来的 flash 播放器。 |

## 效果图

(无)

## 备注

(无)