**iOS12新特性**

**1、性能提升**

**（1）iOS12系统中速度提升**

        应用启动速度提升40%

        键盘响应速度提升50%

        相机启动速度提升70%

在iOS12中对自动布局进行了很大的优化，如图1.1，是官方测试后得到的 iOS 12 和 iOS 11 在特定场景下时间开销的对比图。

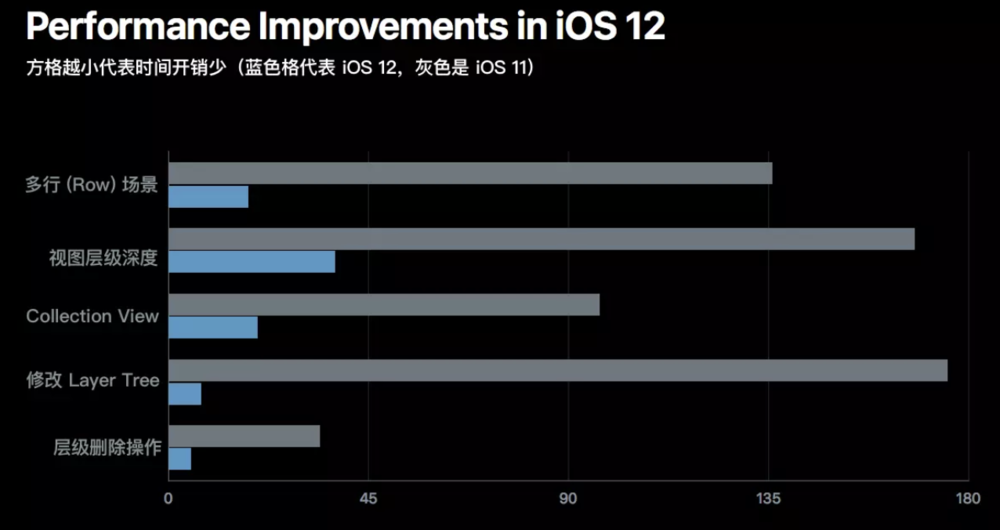


图1.1 iOS11和iOS12下Auto Layout性能对比图

**2、功能迭代**

**（1）UIWebView弃用**

在iOS12 中，UIWebView不在被使用

为什么**？**

原因就是：iOS平台跨域访问漏洞，而造成平台跨域访问漏洞的罪魁祸首就是 UIWebView。



**（2）C++ 标准库libstdc++相关的3个库(libstdc++、libstdc++.6、libstdc++6.0.9 ）废弃**

**使用**libc++代替

***为什么？***

苹果的解释是libstdc++已经标记为废弃有5年了，建议大家使用经过了llvm优化过并且全面支持C++11的libc++库。

**（3）userNotifications**

在iOS12中通知提供了以下五方面的新特性：

        Grouped notifications ：推送分组

         Notification content extensions： 推送内容扩展中的可交互和动态更改Action

        Notification management ：推送消息的管理

        Provisional authorization： 临时授权

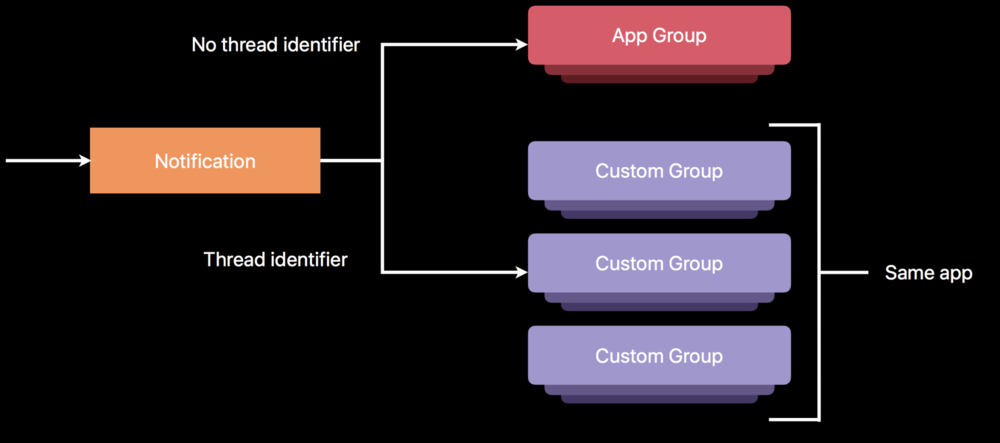
        Critical alerts ：重要提醒

**1）.Grouped notifications**

在iOS12中， 通知分组使用两种分组方式：自动分组(Automatic grouping)和线程标识(Thread identifier).

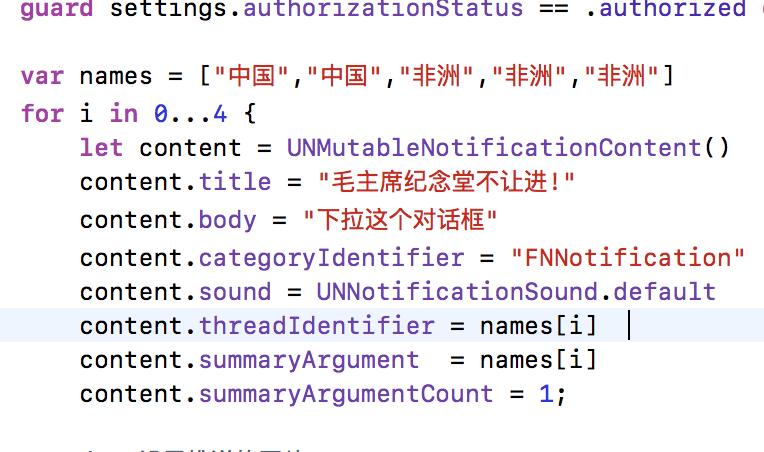
自动分组不需要做额外的操作，系统会根据App的bundle id对推送进行分组。如果需要对通知做更细致的分组就需要用线程标识, 也就是下面的threadIdentifier.

如图



Grouped notifications 结构图

代码和实现效果，如图：



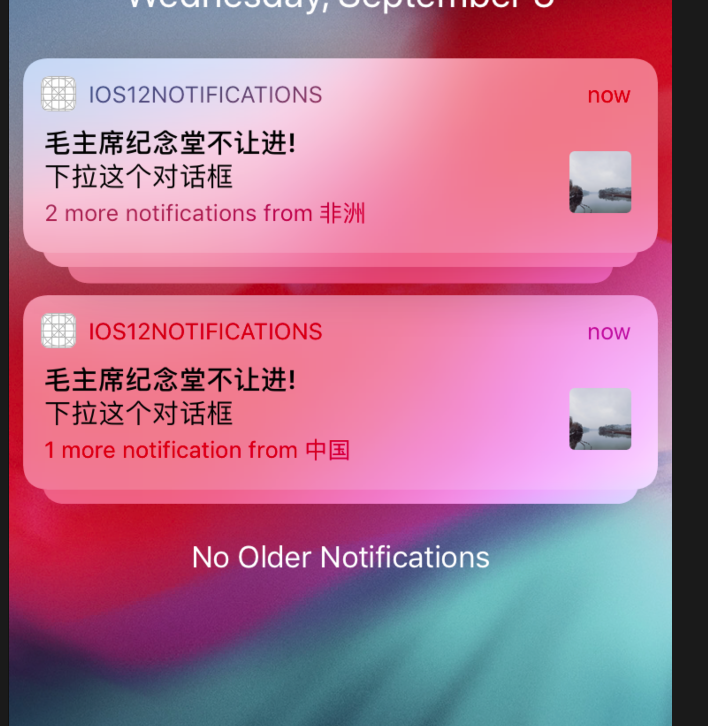


图2.3 通知分组demo实现效果图

同时，在iOS12系统的手机中，可通过 setting -应用 - Notifications - Notification Grouping来设置通知的分组模式。如图2.3所示

***注意：只有设为 Automatic时，设置Thread identifier属性才有效。***

 同时，iOS12 支持Summary 定制。

当苹果自动将推送消息的归拢到一起的时候，最下边会有一个消息摘要。默认格式是:n more notifications from xxx。如图2.4所示

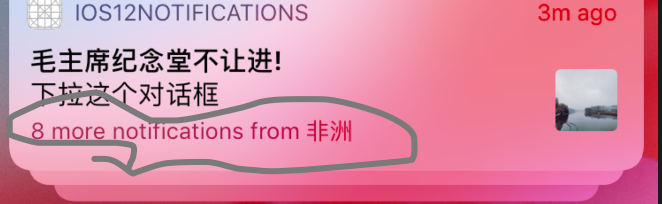


图2.4 Summary（模拟器为例）

**修改方式**： 设置categorySummaryFormat, 会显示summaryFormat, 使用带categorySummaryFormat的init方法,如图2.5所示：



图2.5 summaryFormat实现summary定制

**数字定制**

有时会出现另一个场景：比如发送了2条推送消息，一条是“你有3个邀请函”，另一条是“你有5个邀请函”。那摘要则会显示你有2更多消息，我们想要的肯定是"你有8个邀请函"。那这种效果怎么显示呢？（summaryArgumentCount）

**2）.推送内容扩展中的可交互和动态更改Action**

配置Notification Content Extention的UUNNotificationExtensionUserInteractionEnabled为YES。

通过File-New-Target中的 Notification Content Extension 来创建NotificationViewController。

**3）.增加了推送消息的管理**

展现管理界面：通知消息左滑 或者 进入详情中右上角的，管理界面如图2.7所示：

Deliver Quietly 控制会不会播放声音；Turn Off 关闭推送；

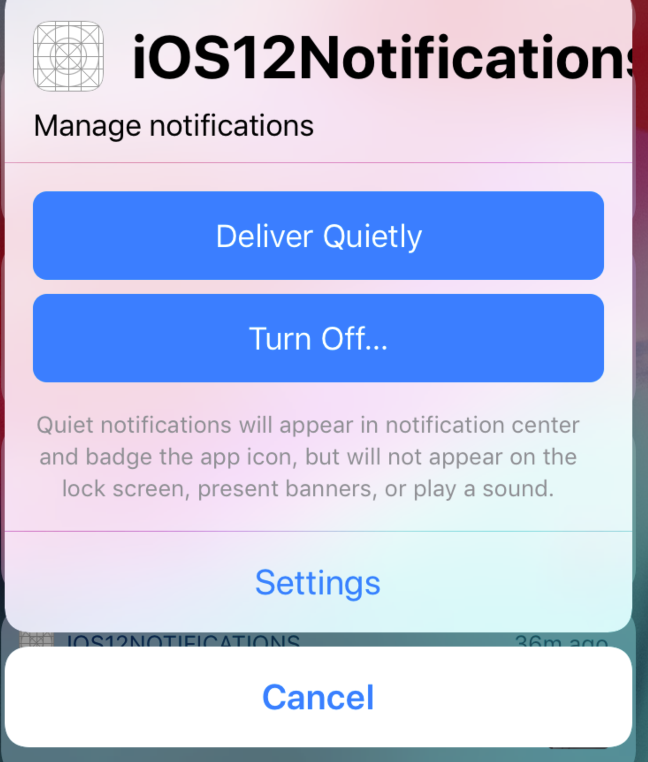


图2.7 通知管理界面

**4）.临时授权（**Provisional Authorization**）**

通知会出现在通知中心并在应用图标上出现标记，但不会显示在锁定屏幕上，不会显示横幅，也不会播放声音,之后可以选择正常推送还是关闭，在申请权限的时候，加上provisional即可

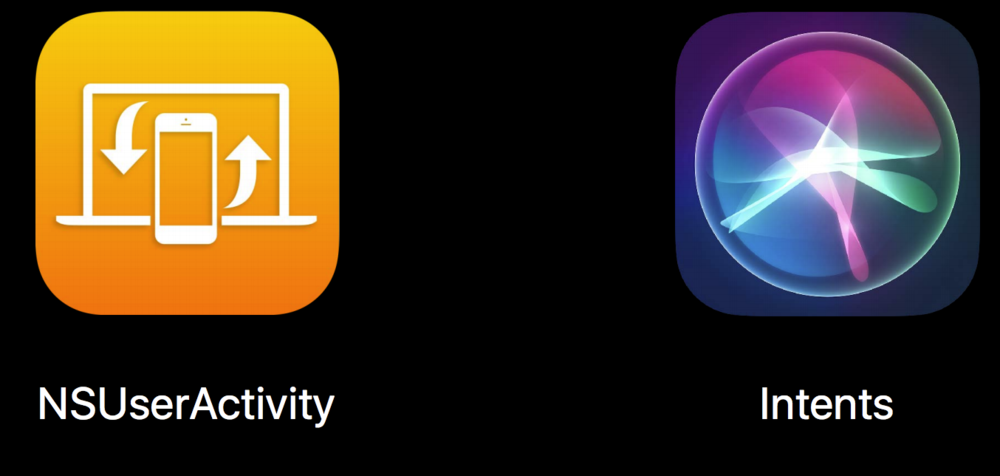
**5）.**重要提醒**（**Critical Alerts**）**

无视勿扰模式和铃声开关的限制，收到这类通知时会伴随一个系统或App设定的提示音 需要获得授权。在申请权限的时候，加上criticalAlert。

**(4) shortcuts**

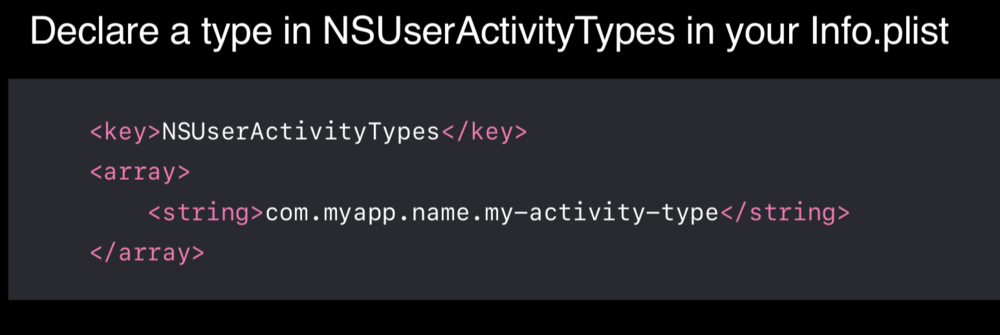
iOS12 提供了语音助手的快捷短语的开发，可以通过自定义一个短语，通过Siri来唤起特定的功能。

Shortcuts APIS

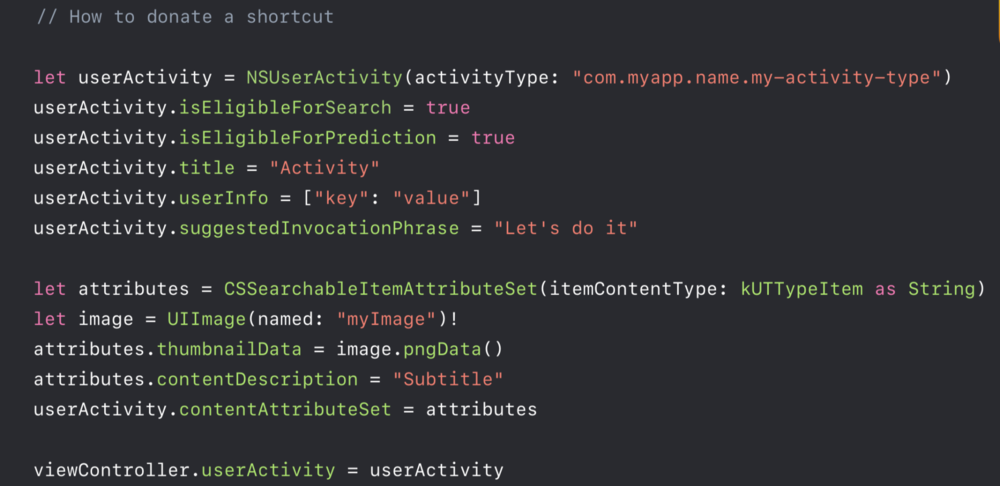


**1）NSUserActivity的使用**

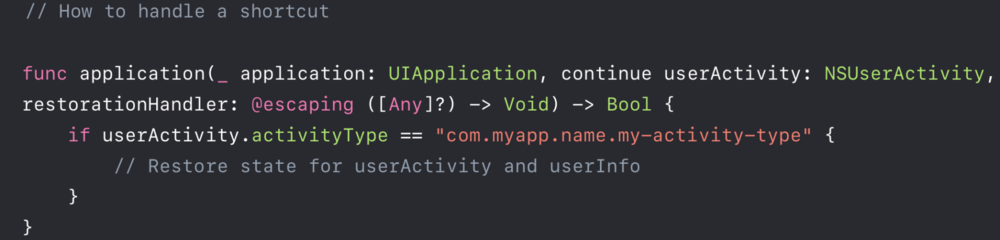
首先，在Info.plist中声明NSUserActivityTypes



然后，生成对应的NSUserActivity



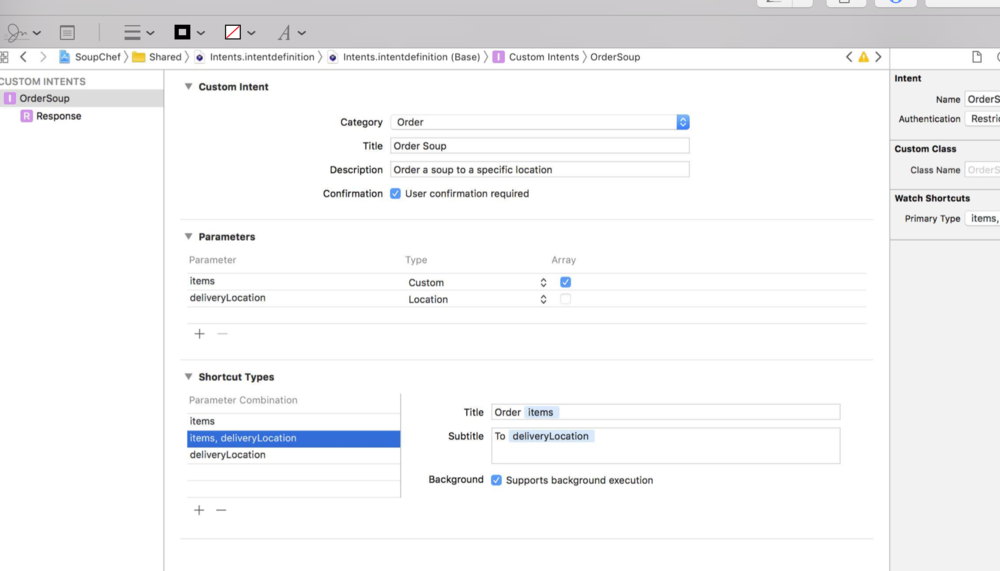
最后，用户点击shortCut 的 处理入口



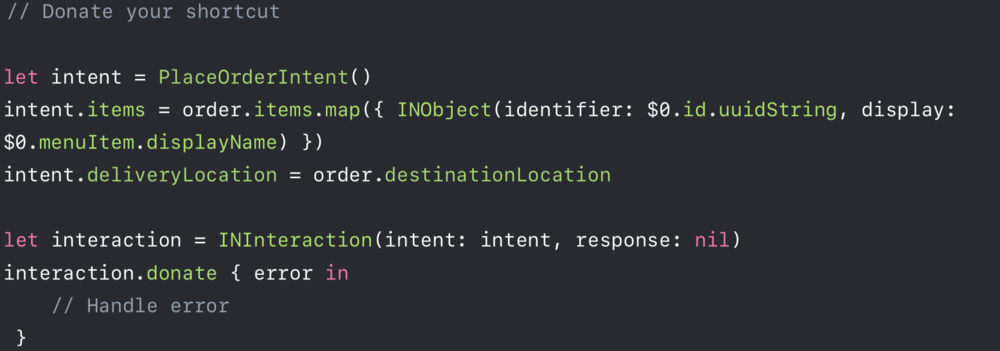
**2）自定义Intents**

通过Intent实现的shortcut是不启动APP不通过交互行为即可以直接运行的功能，并且可以对运行的结果给出响应反馈。在工程中新建一个SiriKit Intent Definition File，点击‘+’新建一个自定义的Intent，然后添加参数。之后在Shortcut Type中添加需要用的参数，并且定义在shortcut中的显示内容

编辑器会通过这个文件自动生成对应的**INIntent**类

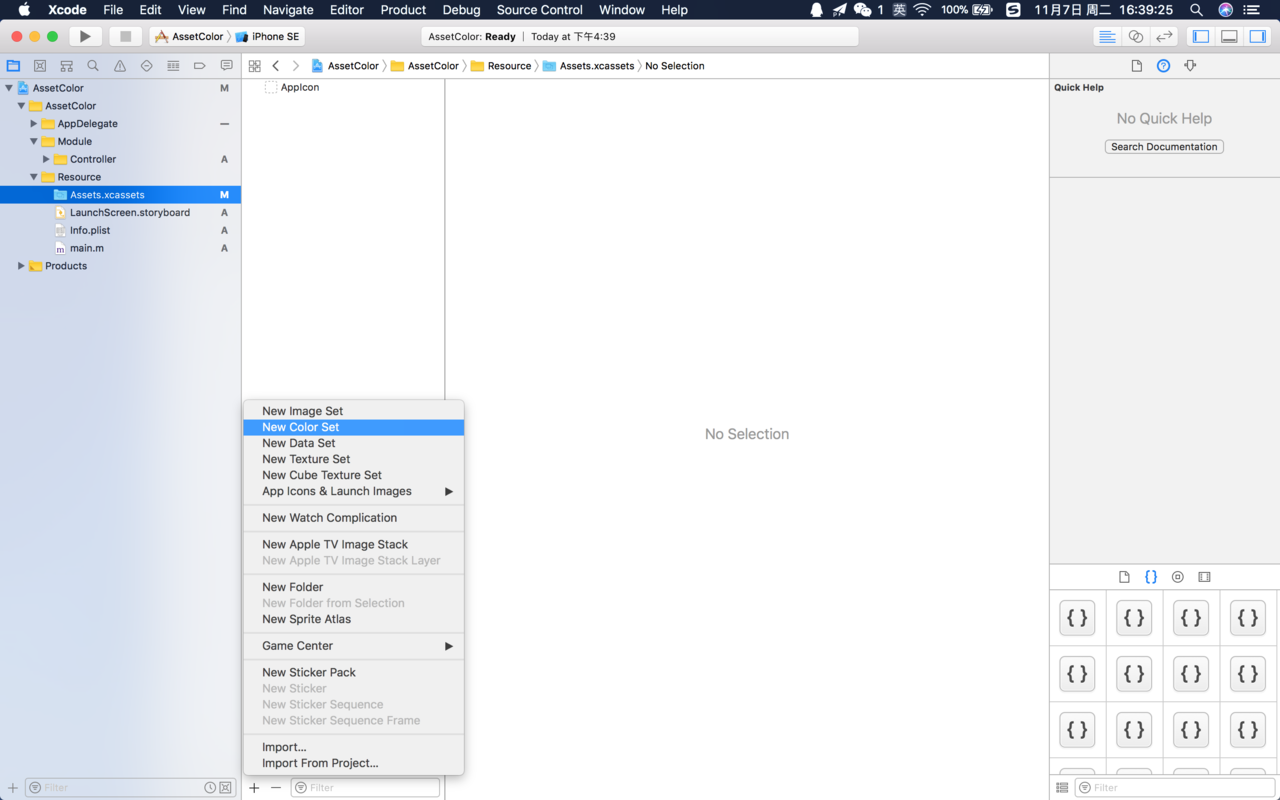


处理入口和NSUserActivity相同。

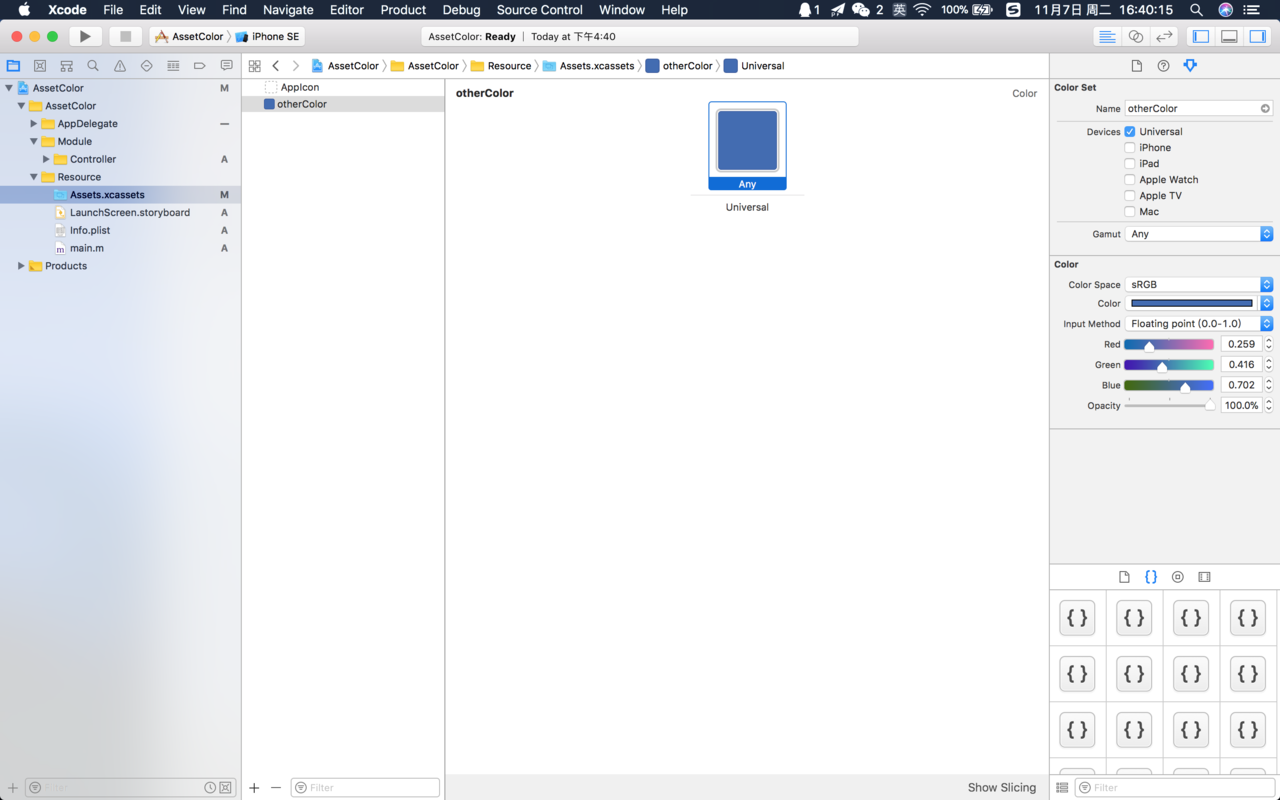


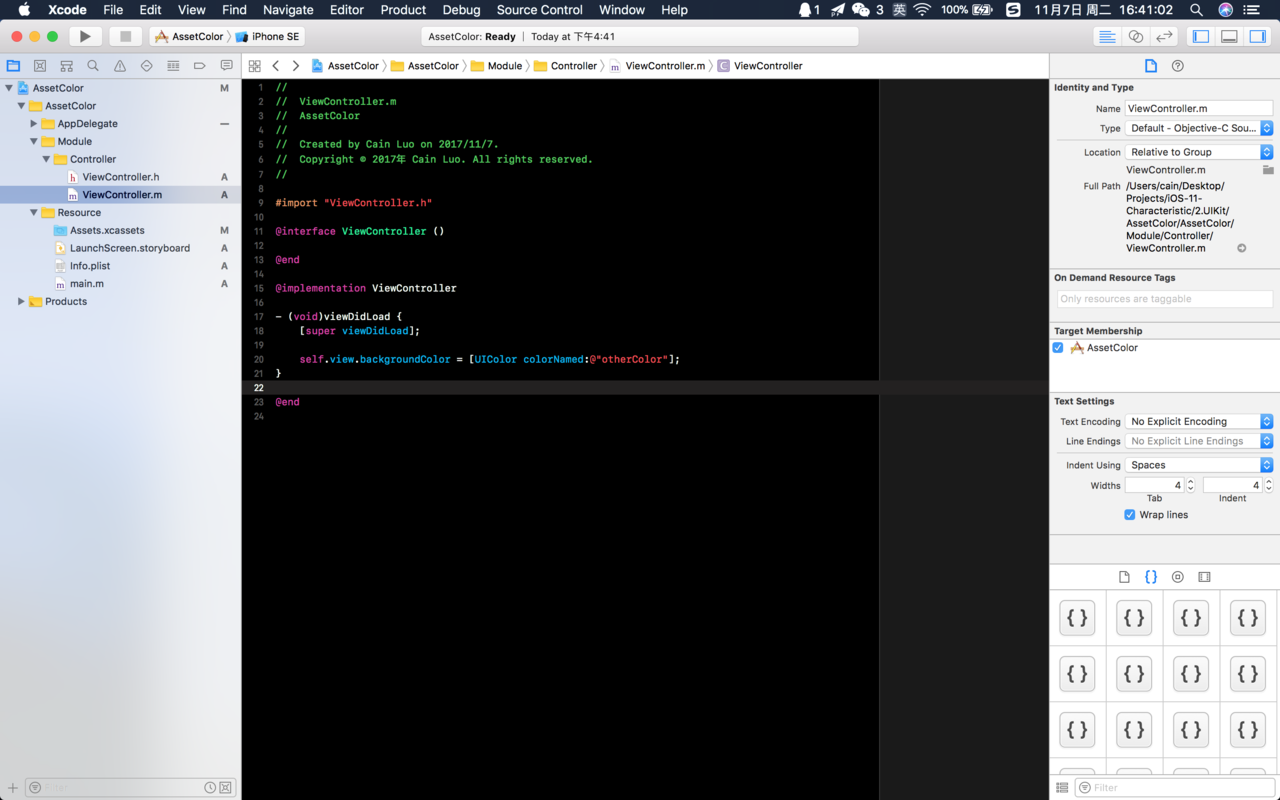
**3、外篇**

1. Asset里可以集成UIColor的目录, 这样子我们就可以省略声明一大堆的颜色, 详细怎么做呢?

然后我们在Assets.xcassets里添加Color Set:

然后添加自己喜欢的颜色值:

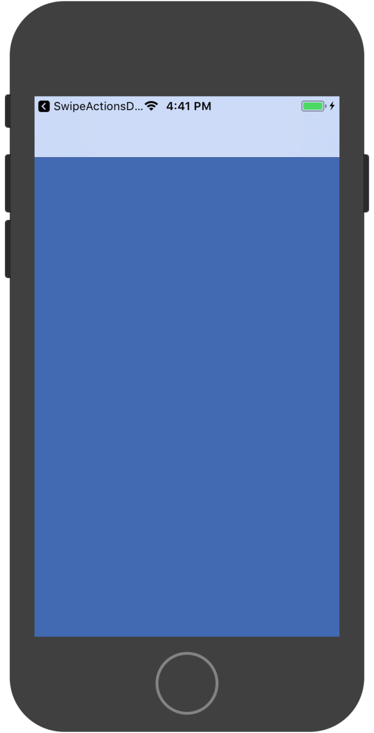




这里我们要看看两个API, 都是在iOS 11之后才出来的

+ (nullable UIColor \*)colorNamed:(NSString \*)name

+ (nullable UIColor \*)colorNamed:(NSString \*)name inBundle:(nullable NSBundle \*)bundle compatibleWithTraitCollection:(nullable UITraitCollection \*)traitCollection



1. 屏幕适配

#define  TabbarHeight         (isFullScreen ? (49.f+34.f) : 49.f)

#define  StatusBarAndNavigationBarHeight  (isFullSc reen ? 88.f : 64.f)