

简
(/)[登录 \(/sign_in\)](/sign_in) [注册 \(/sign_up\)](/sign_up)

(/collections)

**作者** Dely (/users/e95705fe04d9) 2016.09.20 15:56*写了8756字，被171人关注，获得了198个喜欢
(/users/e95705fe04d9)[+ 添加关注 \(/sign_in\)](/sign_in)

(/apps)

iOS 10 适配知识点总结


字数3307 阅读6692 评论33 喜欢69

背景

刚刚过完中秋节，第二天app上线被拒，原因是因为启动app就会crash，领导大早上给我打电话让我去公司解决，好吧谁让人家是领导呢！正好iOS10系统刚刚出来，需要适配iOS10，不然上线还是会拒，所以我果断升级了xcode8.0。我总结了一些资料，接下来介绍一下iOS适配的一些知识点：

1.证书问题

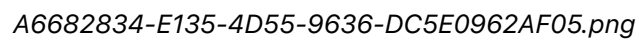
- 打开xcode8.0时编译运行时出现下面问题：

39B1F3AE-530A-46B7-BE6E-379A0AA6B4DE.png

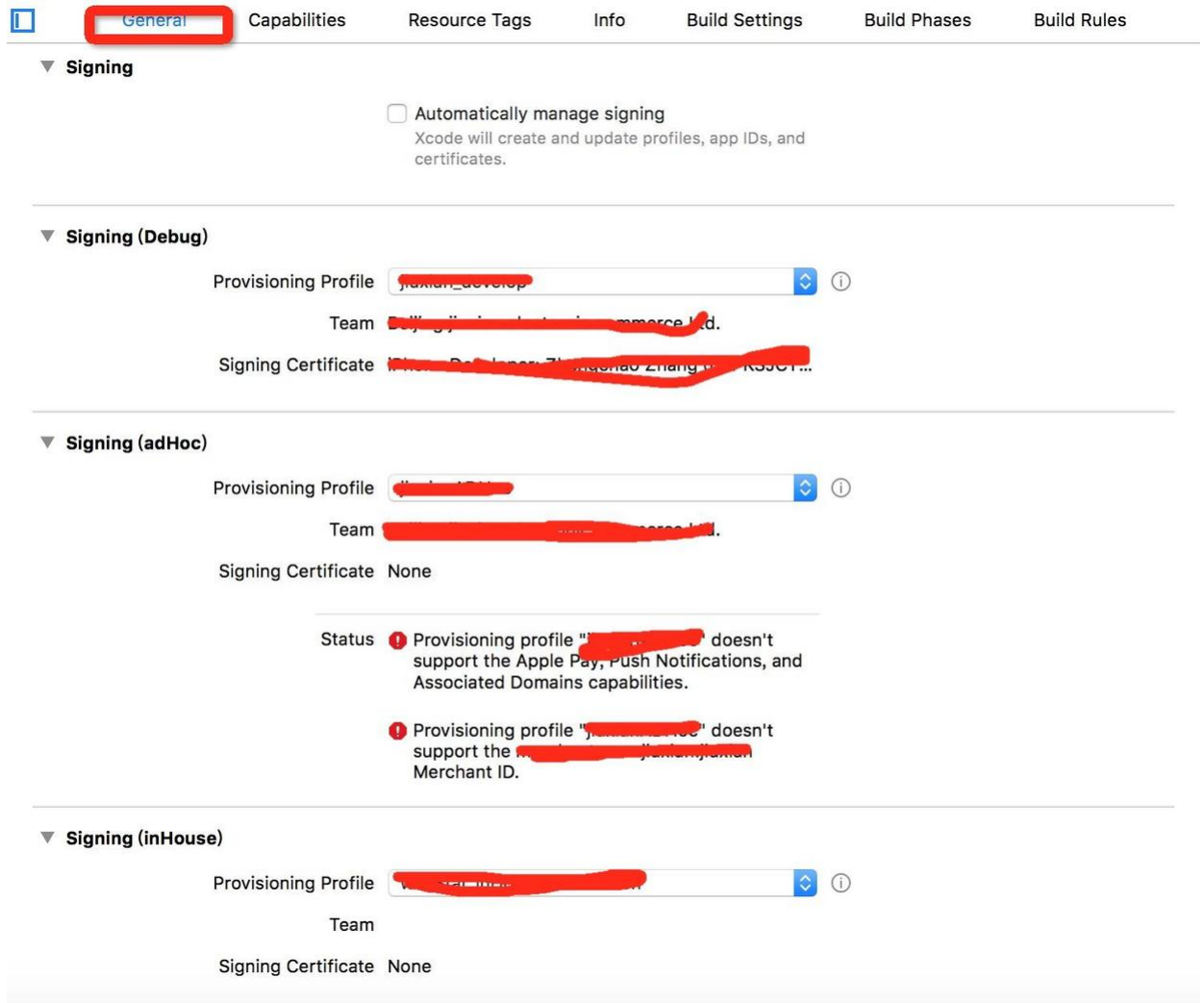
这个问题刚开始估计大家都会碰到也是第一个要解决的问题

这个问题就是一个证书的设置问题，下面看两张图

正常我们会在BuildSettings中设置证书：



<http://www.jianshu.com/p/61b73ac8662a>



06300540-83C7-44F4-9069-F198599F0086.png

相信大家都能看到在General下面会有Signing，没错这就是新特性，为了方便用户来管理，大家可以选择**Automatically manage signing**。需要输入开发者账号！如果没有账号也没关系，在下面也可以选择**Debug**、**Release**、**inHouse**模式下对应的证书也可以！

- 但是 但是 但是 如果你的证书如果是通配符类型的，但是你的app包含了比如**推送**、**apple pay**、他会报错提示你未报含xx.id的manteid等等。所以你要生成针对你app的bundleid对应的证书！

2.隐私数据访问问题

问题出现

- 现在app能运行了，当我打开相机时突然又crash了，好吧，坑还不少，崩溃日志如

下:

This app has crashed because it attempted to access privacy-sensitive data without a usage description. The app's Info.plist must contain an NSPhotoLibraryUsageDescription key with a string value explaining to the user how the app uses this data.

崩溃原因

- 上面崩溃意思试图访问安全隐私数据，但是又没有相应的描述，你必须要包含在**info.plist**里包含一个键值 **NSPhotoLibraryUsageDescription**对应的描述。
- 其实上面是iOS10，苹果加强了对隐私数据的保护，要对隐私数据权限做一个适配 iOS10调用相机，访问通讯录，访问相册等都要在**info.plist**中加入权限访问描述，不然之前你们的项目涉及到这些权限的地方就会直接crash掉。

解决办法

- 只需要在info.plist添加NSContactsUsageDescription的key, value自己随意填写就可以,这里列举出对应的key(Source Code模式下):

```
<key>NSPhotoLibraryUsageDescription</key>
<string>App需要您的同意,才能访问相册</string>

<key>NSCameraUsageDescription</key>
<string>App需要您的同意,才能访问相机</string>

<key>NSMicrophoneUsageDescription</key>
<string>App需要您的同意,才能访问麦克风</string>

<key>NSLocationUsageDescription</key>
<string>App需要您的同意,才能访问位置</string>

<key>NSLocationWhenInUseUsageDescription</key>
<string>App需要您的同意,才能在使用期间访问位置</string>

<key>NSLocationAlwaysUsageDescription</key>
<string>App需要您的同意,才能始终访问位置</string>

<key>NSCalendarsUsageDescription</key>
<string>App需要您的同意,才能访问日历</string>

<key>NSRemindersUsageDescription</key>
<string>App需要您的同意,才能访问提醒事项</string>

<key>NSMotionUsageDescription</key>
<string>App需要您的同意,才能访问运动与健身</string>

<key>NSHealthUpdateUsageDescription</key>
<string>App需要您的同意,才能访问健康更新 </string>

<key>NSHealthShareUsageDescription</key>
<string>App需要您的同意,才能访问健康分享</string>

<key>NSBluetoothPeripheralUsageDescription</key>
<string>App需要您的同意,才能访问蓝牙</string>

<key>NSAppleMusicUsageDescription</key>
<string>App需要您的同意,才能访问媒体资料库</string>
```

- 隐私数据	对应key值
相册	NSPhotoLibraryUsageDescription
相机	NSCameraUsageDescription
麦克风	NSMicrophoneUsageDescription
位置	NSLocationUsageDescription

在使用期间访问位置	NSLocationWhenInUseUsageDescription
始终访问位置	NSLocationAlwaysUsageDescription
日历	NSCalendarsUsageDescription
提醒事项	NSRemindersUsageDescription
运动与健身	NSMotionUsageDescription
健康更新	NSHealthUpdateUsageDescription
健康分享	NSHealthShareUsageDescription
蓝牙	NSBluetoothPeripheralUsageDescription
媒体资料库	NSAppleMusicUsageDescription

跳转到app内的隐私数据设置页面

- 我们知道用户没开启，肯定要提醒用户去设置开启的吧，所以我们要跳转到app的隐私数据界面。如何调用呢，我定义一个宏方便调用：

```
UIKit_EXTERN      NSString      *const      UIApplicationOpenSettingsURLString
NS_AVAILABLE_IOS(8_0);
这个是iOS8.0以后的方法
```

```
//-----系统权限设置路径( iOS8以后适用)-----
//url
#define SettingURL [NSURL URLWithString:UIApplicationOpenSettingsURLString]
//调到设置
#define GoToSetting if([[UIApplication sharedApplication] canOpenURL:SettingURL]) {
                    [[UIApplication sharedApplication] openURL:SettingURL];} }
```

我们直接调用下面代码就可以了

```
//去设置
if (IOS8_OR_LATER) {
    GoToSetting;
}
```

看到评论说跳到蓝牙有什么办法，我补充一下：

1. 首先你在info.plist的文件中添加相应权限
2. 在要跳转蓝牙设置界面添加类#import <CoreBluetooth/CoreBluetooth.h> 并声明代理CBCentralManager
3. 设置全局的控件@property (nonatomic, strong)CBCentralManager *testCB;
4. 在要跳转的地方写(最好不要写局部变量, 因为你得拿到这个CB来做连接设备等一些处理。当然你要局部变量也没

```
if (_testCB) {  
    _testCB = nil;  
    _testCB.delegate = nil;  
    _testCB = [[CBCentralManager alloc] initWithDelegate:self queue:nil];  
}else{  
    _testCB = [[CBCentralManager alloc] initWithDelegate:self queue:nil];  
}
```

5. 若要获得蓝牙状态, 实现代理方法

```
-(void)centralManagerDidUpdateState:(CBCentralManager *)central{  
}
```

这样如果蓝牙关闭会提示"打开蓝牙来允许"xxx"连接到配件", 这样点击设置就可以跳到蓝牙设置界面了



A55064BB-2BD5-4DEA-A733-862761D76F5B.png



88F76B47-A82A-4426-A5AE-8EB42CAB0DEA.png

iOS 10 干掉了所有系统设置的 URL Scheme，这意味着你再也不可能直接跳转到系统设置页面（比如 WiFi、蜂窝数据、定位等）。

iOS 10中如下跳到系统的设置方法已经不生效了(如果看到解决办法再补充，如果你有解决办法请赐教留下你宝贵的评论.....感谢):

```
//代码失效,谨慎使用
[[UIApplication sharedApplication] openURL:[NSURL URLWithString:@"prefs:root=WIFI"]]
```

3.系统版本判断方法失效

- 我们之前的系统版本方法如下
- 当系统版本到iOS10.0的时候 9.0和10.0比较的话是降序而不是升序，这样会导致iOS10.0是最早的版本，这样后面要走的iOS10的方法可能都不会走而出现问题

```
#define IOS9_OR_LATER ([[UIDevice currentDevice] systemVersion] compare:@"9.0"] != NSOrderedAscending)

#define IOS8_OR_LATER ([[UIDevice currentDevice] systemVersion] compare:@"8.0"] != NSOrderedAscending)

#define IOS7_OR_LATER ([[UIDevice currentDevice] systemVersion] compare:@"7.0"] != NSOrderedAscending)

#define IOS6_OR_LATER ([[UIDevice currentDevice] systemVersion] compare:@"6.0"] != NSOrderedAscending)
```


- 下面这样也不行它会永远返回NO,substringToIndex:1在iOS 10 会被检测成 iOS 1了,

```
#define isiOS10 ([[[[UIDevice currentDevice] systemVersion] substringToIndex:1] intValue]
```

- 正确的打开方式应该是:

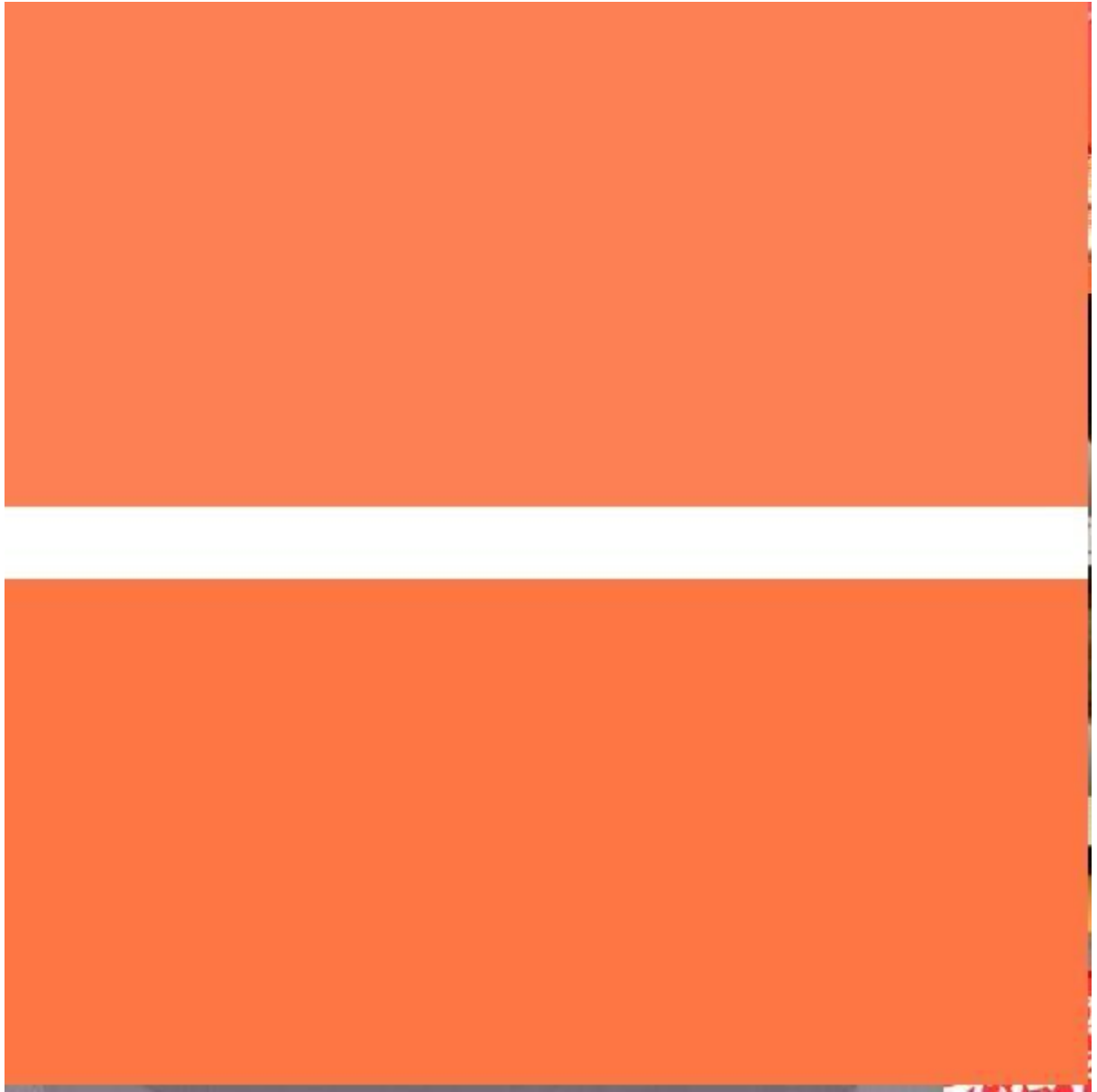
```
#define IOS10_OR_LATER ([[[[UIDevice currentDevice] systemVersion] floatValue] >= 10.0)
#define IOS9_OR_LATER ([[[[UIDevice currentDevice] systemVersion] floatValue] >= 9.0)
#define IOS8_OR_LATER ([[[[UIDevice currentDevice] systemVersion] floatValue] >= 8.0)
#define IOS7_OR_LATER ([[[[UIDevice currentDevice] systemVersion] floatValue] >= 7.0)
#define IOS6_OR_LATER ([[[[UIDevice currentDevice] systemVersion] floatValue] >= 6.0)
```

4.UIColor问题

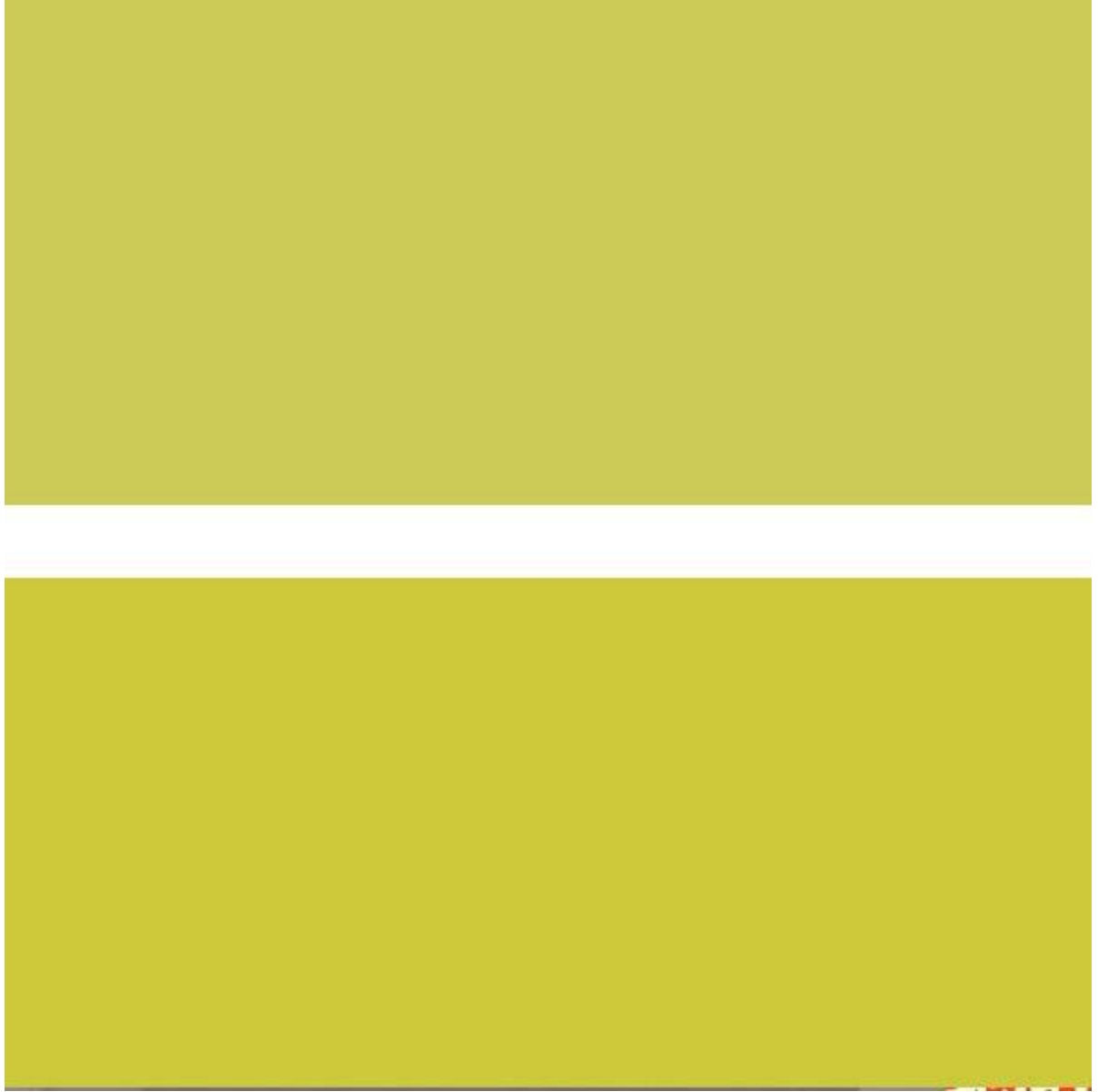
- 官方文档中说:大多数core开头的图形框架和AVFoundation都提高了对扩展像素和宽色域色彩空间的支持.通过图形堆栈扩展这种方式比以往支持广色域的显示设备更加容易。现在对UIKit扩展可以在sRGB的色彩空间下工作，性能更好,也可以在更广泛的色域来搭配sRGB颜色.如果你的项目中是通过低级别的api自己实现图形处理的,建议使用sRGB,也就是说在项目中使用了RGB转化颜色的建议转换为使用sRGB,在UIColor类中新增了两个api:

```
+ (UIColor *)colorWithDisplayP3Red:(CGFloat)displayP3Red green:(CGFloat)green blue:(CGFloat)blue
- (UIColor *)initWithDisplayP3Red:(CGFloat)displayP3Red green:(CGFloat)green blue:(CGFloat)blue
```

我用新老方法测试两个方法在RGB相同的数值在表现上的区别看下图:



06E01411-216F-4BE6-B36D-27FE3FEA790D.png



F354DBB8-6192-403B-9817-024257DE1310.png

- 可以看出下面的颜色（sRGB方法）比上面的颜色（RGB方法）颜色更深更明显。

5.真彩色的显示

- 真彩色的显示会根据光感应器来自动的调节达到特定环境下显示与性能的平衡效果, 如果需要这个功能的话,可以在info.plist里配置(在Source Code模式下):

```
<key>UIWhitePointAdaptivityStyle</key>
```

它有五种取值,分别是:

```
<string>UIWhitePointAdaptivityStyleStandard</string> // 标准模式
<string>UIWhitePointAdaptivityStyleReading</string> // 阅读模式
<string>UIWhitePointAdaptivityStylePhoto</string> // 图片模式
<string>UIWhitePointAdaptivityStyleVideo</string> // 视频模式
<string>UIWhitePointAdaptivityStyleStandard</string> // 游戏模式
```

- 如果你的项目是游戏类的,就选择UIWhitePointAdaptivityStyleStandard这个模式,五种模式的显示效果是从上往下递减,也就是说如果你的项目是图片处理类的,你选择的是阅读模式,给选择太好的效果会影响性能.

6.字体变化

- 苹果的默认字体随着iOS系统版本的不同而不同, iOS10中字体变大了。导致了原来的显示有问题, 会造成...的出现。暂时没有好的解决办法, 需要自己在一个个适配一下!

7.插件取消

- Xcode8取消了三方插件的功能, 好多教程破解可以继续使用, 但是可能app上线可能会被拒。我们最喜爱的VVDocumenter-Xcode也不能使用了, 下面是作者的感谢

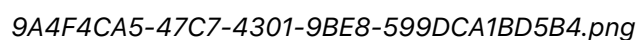
VVDocumenter-Xcode  

Goodbye World

In Xcode 8, Apple integrated a comment documentation generator plugin, which is built on top of VVDocumenter. Now this project is proud to be a part of Apple. It means you could just use the shortcut (`⌘ Option + ⌘ Command + /`) to add a documentation comment to your code if you are using Xcode 8 or above!

VVDocumenter is one of my hobby projects [from 2013](#), back to the age of Xcode 4. It serves well for these years and I am so glad that it helps a lot of developers to improve their productivity. Since there is no need to install this plugin anymore, the development of VVDocumenter will not continue. Yes, it's time to say goodbye, with a happy ending.

Thank you all for your selfless support to this project. Let's build more great things and make the world better in future!

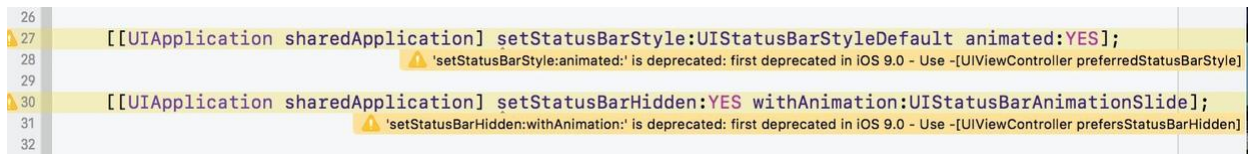


看来大神都是谦虚的啊(啥时候能成为大神。我还是洗洗睡吧,梦里啥都有\^_^)

- 上面也提到了我们可以继续使用注释, 快捷键 (`⌘ Option + ⌘ Command + /`)

8.UIStatusBar的问题

- 在iOS10中,如果还使用以前设置UIStatusBar类型或者控制隐藏还是显示的方法,会报警告,方法过期,如下图:



21A56CDA-7A82-433C-B446-24C428087D1F.png

警告中提到从iOS9.0开始就弃用这两个方法了, 需要用

`-[UIViewController preferredStatusBarStyle]`

`-[UIViewController preferredStatusBarHidden]`来替换使用, 那我们来看看新的替换方法。

- 新技能见下面

```
#if UIKIT_DEFINE_AS_PROPERTIES
@property(nonatomic, readonly) UIStatusBarStyle preferredStatusBarStyle NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED;
@property(nonatomic, readonly) BOOL prefersStatusBarHidden NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED; // Defaults
// Override to return the type of animation that should be used for status bar change
@property(nonatomic, readonly) UIStatusBarAnimation preferredStatusBarUpdateAnimation NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED;
#else
- (UIStatusBarStyle)preferredStatusBarStyle NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED;
- (BOOL)prefersStatusBarHidden NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED; // Defaults
// Override to return the type of animation that should be used for status bar change
- (UIStatusBarAnimation)preferredStatusBarUpdateAnimation NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED;
#endif
```

- 上面这个新方法在UIViewController.h文件中, 这说明什么? 当然说明这是 **viewController** 的属性和方法了, 只需要在viewController里调用修改即可
- UIStatusBarStyle** 和 **prefersStatusBarHidden** 这两个属性是 **readonly** **readonly** 也就是说我们如果调用下面 肯定是报错的:

```
//这是错误的写法
self.preferredStatusBarStyle = UIStatusBarStyleDefault;和
self.prefersStatusBarHidden = YES;
```

- 正确的打开方式在viewController重写我们还没用的新的方法

```
//这是正确的
- (BOOL)prefersStatusBarHidden{
    return YES;
}

- (UIStatusBarStyle)preferredStatusBarStyle{
    return UIStatusBarStyleDefault;
}
```

9.UITextField(好像作用并不大)

- 在iOS 10 中,UITextField新增了textContentType字段,是UITextContentType类型,它是一个枚举,作用是可以指定输入框的类型,以便系统可以分析出用户的语义.是电话类型就建议一些电话,是地址类型就建议一些地址.可以在#import <UIKit/UITextInputTraits.h>文件中,查看textContentType字段,有以下可以选择的类型:

```
UIKIT_EXTERN UITextContentType const UITextContentTypeName NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeNamePrefix NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeGivenName NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeMiddleName NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeFamilyName NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeNameSuffix NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeNickname NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeJobTitle NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeOrganizationName NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeLocation NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeFullStreetAddress NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeStreetAddressLine1 NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeStreetAddressLine2 NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeAddressCity NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeAddressState NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeAddressCityAndState NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeSublocality NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeCountryName NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypePostalCode NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeTelephoneNumber NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeEmailAddress NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeURL NS_A
UIKIT_EXTERN UITextContentType const UITextContentTypeCreditCardNumber NS_A
```

10.UICollectionViewCell的优化

- 在iOS 10 之前, UICollectionView上面如果有大量cell, 当用户活动很快的时候, 整个UICollectionView的卡顿会很明显, 为什么会造成这样的问题, 这里涉及到了iOS 系统的重用机制, 当cell准备加载进屏幕的时候, 整个cell都已经加载完成, 等待在屏幕外面了, 也就是整整一行cell都已经加载完毕, 这就是造成卡顿的主要原因, 专业术语叫做: 掉帧. 要想让用户感觉不到卡顿, 我们的app必须帧率达到60帧/秒, 也就是说每帧16毫秒要刷新一次.
- iOS 10 之前UICollectionViewCell的生命周期是这样的:
 - 用户滑动屏幕, 屏幕外有一个cell准备加载进来, 把cell从reusr队列拿出来, 然后调用prepareForReuse方法, 在这个方法里面, 可以重置cell的状态, 加载新的数据;
 - 继续滑动, 就会调用cellForItemAtIndexPath方法, 在这个方法里面给cell赋值模型, 然后返回给系统;
 - 当cell马上进去屏幕的时候, 就会调用willDisplayCell方法, 在这个方法里面我们还可以修改cell, 为进入屏幕做最后的准备工作;
 - 执行完willDisplayCell方法后, cell就进去屏幕了. 当cell完全离开屏幕以后, 会调用didEndDisplayingCell方法.
- iOS 10 UICollectionViewCell的生命周期是这样的:
 - 用户滑动屏幕, 屏幕外有一个cell准备加载进来, 把cell从reusr队列拿出来, 然后调用prepareForReuse方法, 在这里当cell还没有进去屏幕的时候, 就已经提前调用这个方法了, 对比之前的区别是之前是cell的上边缘马上进去屏幕的时候就会调用该方法, 而iOS 10 提前到cell还在屏幕外面的时候就调用;
 - 在cellForItemAtIndexPath中创建cell, 填充数据, 刷新状态等操作, 相比于之前也提前了;
 - 用户继续滑动的话, 当cell马上就需要显示的时候我们再调用willDisplayCell方法, 原则就是: 何时需要显示, 何时再去调用willDisplayCell方法;
 - 当cell完全离开屏幕以后, 会调用didEndDisplayingCell方法, 跟之前一样, cell会进入

重用队列.

- 在 iOS 10 之前,cell 只能从重用队列里面取出,再走一遍生命周期,并调用 `cellForItemAtIndexPath` 创建或者生成一个 cell.
- 在 iOS 10 中,系统会 cell 保存一段时间,也就是说当用户把 cell 滑出屏幕以后,如果又滑动回来,cell 不用再走一遍生命周期了,只需要调用 `willDisplayCell` 方法就可以重新出现在屏幕中了.
- iOS 10 中,系统是一个一个加载 cell 的,二以前是一行一行加载的,这样就可以提升很多性能;
- iOS 10 新增加的 Pre-Fetching 预加载

这个是为了降低 `UICollectionViewCell` 在加载的时候所花费的时间,在 iOS 10 中,除了数据源协议和代理协议外,新增加了一个 **`UICollectionViewDataSourcePrefetching`** 协议,这个协议里面定义了两个方法:

```
- (void)collectionView:(UICollectionView *)collectionView prefetchItemsAtIndexPaths:  
- (void)collectionView:(UICollectionView *)collectionView cancelPrefetchingForItemsA
```

- 在 `CollectionView prefetchItemsAt indexPaths` 这个方法是异步预加载数据的,当中的 `indexPaths` 数组是有序的,就是 item 接收数据的顺序;
`CollectionView cancelPrefetchingForItemsAt indexPaths` 这个方法是可选的,可以用来处理在滑动中取消或者降低提前加载数据的优先级.
注意:这个协议并不能代替之前读取数据的方法,仅仅是辅助加载数据.
Pre-Fetching 预加载对 `UITableViewCell` 同样适用.

11.UIRefreshControl

- 在 iOS 10 中, `UIRefreshControl` 可以直接在 `UICollectionView` 和 `UITableView` 中使用,并且脱离了 `UITableViewController`. 现在 `RefreshControl` 是 `UIScrollView` 的一个属性.
使用方法:


```
//创建
UIRefreshControl *refreshControl = [[UIRefreshControl alloc] init];
refreshControl.tintColor = [UIColor redColor];
refreshControl.attributedTitle = [[NSAttributedString alloc] initWithString:@"正在刷新"];
[refreshControl addTarget:self action:@selector(loadData) forControlEvents:UIControlEventTouchUpInside];

//开始和停止刷新
[refreshControl beginRefreshing];
[refreshControl endRefreshing];
```

- 也可以进去头文件查看

```
#import<UIRefreshControl.h>

- (instancetype)init;

@property (nonatomic, readonly, getter=isRefreshing) BOOL refreshing;

@property (nonatomic, resettable, nonatomic, strong) UIColor *tintColor;
@property (nonatomic, nullable, nonatomic, strong) NSAttributedString *attributedTitle UI_APPEARANCE_SELECTOR;

// May be used to indicate to the refreshControl that an external event has initiated a refresh
- (void)beginRefreshing NS_AVAILABLE_IOS(6_0);
// Must be explicitly called when the refreshing has completed
- (void)endRefreshing NS_AVAILABLE_IOS(6_0);
```



12.UserNotifications(用户通知)

- iOS 10 中将通知相关的 API 都统一了,苹果对这是做了重大改进,变的非常易用。

iOS 9 以前的通知

在调用方法时，有些方法让人很难区分，容易写错方法，这让开发者有时候很苦恼。
应用在运行时和非运行时捕获通知的路径还不一致。

应用在前台时，是无法直接显示远程通知，还需要进一步处理。

已经发出的通知是不能更新的，内容发出时是不能改变的，并且只有简单文本展示方式，扩展性根本不是很好。

iOS 10 开始的通知

所有相关通知被统一到到了UserNotifications.framework框架中。

增加了撤销、更新、中途还可以修改通知的内容。

通知不在是简单的文本了，可以加入视频、图片，自定义通知的展示等等。

iOS 10相对之前的通知来说更加好用易于管理，并且进行了大规模优化，对于开发者来说是一件好事。

iOS 10开始对于权限问题进行了优化，申请权限就比较简单了(本地与远程通知集成在一个方法中)。

iOS 10 通知学习相关资料：

1. UserNotifications: 苹果官方文档
(<https://developer.apple.com/reference/usernotifications>)
2. 活久见的重构 - iOS 10 UserNotifications 框架解析
(<https://onevcat.com/2016/08/notification/>)
3. WWDC2016 Session 笔记 - iOS 10 推送 Notification 新特性
(<http://www.jianshu.com/p/9b720efe3779>)

后面对UserNotifications单独发表文章学习相关的知识

参考资料：

1. <http://blog.csdn.net/jiang314/article/details/52502450>
(<http://blog.csdn.net/jiang314/article/details/52502450>)
2. <http://www.jianshu.com/p/90d5323cf510>

(<http://www.jianshu.com/p/90d5323cf510>)

3. 个人博客 (<http://dely.vip>) <http://dely.vip> (<http://dely.vip>)

➕ 推荐拓展阅读 (/sign_in)

© 著作权归作者所有

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

¥ 打赏支持



(/users/b01e93d8ff2e)

♡ 喜欢 | 69

分享到微博 分享到微信
更多分享 ▼

33条评论 (按时间正序 · 按时间倒序 · 按喜欢排序)

添加新评论 (/sign_in)



梦想总是美好的 (/users/eae44b804245)

2/楼 2016.09.20 16:04 (/p/61b73ac8662a/comments/4303316#comment-4303316)

😊😊😊 thanks for your work .

♡ 喜欢(1)

回复

Dely (/users/e95705fe04d9): @梦想总是美好的 (/users/eae44b804245) thanks for your support 😊
2016.09.20 16:23 (/p/61b73ac8662a/comments/4303619#comment-4303619)

回复

添加新回复



fulen (/users/23622bffca02)

3/楼 2016.09.20 22:30 (/p/61b73ac8662a/comments/4310887#comment-4310887)

😊😊😊 thanks for your work .

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @23622bffca02 (/users/23622bffca02) 谢谢 😊

2016.09.21 09:28 (/p/61b73ac8662a/comments/4316600#comment-4316600)

回复

✎ 添加新回复



R_Alex (/users/e7e31e11fef2)

4楼 2016.09.21 06:32 (/p/61b73ac8662a/comments/4314246#comment-4314246)

感谢分享

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @R_Alex (/users/e7e31e11fef2) 谢谢支持~ 😊

2016.09.21 09:29 (/p/61b73ac8662a/comments/4316618#comment-4316618)

回复

✎ 添加新回复



永不言败1 (/users/5e39f65c38ad)

5楼 2016.09.22 11:34 (/p/61b73ac8662a/comments/4340759#comment-4340759)

😊😊😊 thanks for your work .

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @永不言败1 (/users/5e39f65c38ad) 😊 thanks for your support.

2016.09.22 16:38 (/p/61b73ac8662a/comments/4346686#comment-4346686)

回复

✎ 添加新回复



那份牵挂给了谁 (/users/95b1a0ecf620)

6楼 2016.09.22 11:37 (/p/61b73ac8662a/comments/4340844#comment-4340844)

UIApplicationOpenSettingsURLString跳过去后没有蓝牙设置，如果要跳到蓝牙设置的话，有什么办法嘛？

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @那份牵挂给了谁 (/users/95b1a0ecf620)

- 1.首先你在info.plist的文件中添加相应权限
- 2.在要跳转蓝牙设置界面添加类#import <CoreBluetooth/CoreBluetooth.h> 并声明代理<CBCentralManagerDelegate>
- 3.设置全局的控件@property (nonatomic, strong)CBCentralManager *testCB;
- 4.在要跳转的地方写
_testCB = [[CBCentralManager alloc] initWithDelegate:self queue:nil];
- 5.若要获得蓝牙状态, 实现代理方法
-(void)centralManagerDidUpdateState:(CBCentralManager *)central{
}

这样如果蓝牙关闭会提示"打开蓝颜来允许"xxx"连接到配件", 这样点击设置就可以跳到蓝牙设置界面了



2016.09.22 16:37 (/p/61b73ac8662a/comments/4346663#comment-4346663)

回复

那份牵挂给了谁 (/users/95b1a0ecf620): @Dely (/users/e95705fe04d9) 这个是系统自带的跳转, 在创建CBCentralManager的时候, 去检测是否开启蓝牙的。如果用户取消了这个提示。怎么办?

2016.09.23 11:47 (/p/61b73ac8662a/comments/4362631#comment-4362631)

回复

Dely (/users/e95705fe04d9): @那份牵挂给了谁 (/users/95b1a0ecf620), 用户取消了, 也没关系, 下次点击如果没有开启蓝牙, 会再次提醒的 😊

2016.09.23 16:46 (/p/61b73ac8662a/comments/4368449#comment-4368449)

回复

还有 2 条评论, 展开查看

✎ 添加新回复



木子同学 (/users/b01e93d8ff2e)

7楼 2016.09.22 18:43 (/p/61b73ac8662a/comments/4348856#comment-4348856)



Thanks for your work!

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @木子同学 (/users/b01e93d8ff2e) 😊😊 thanks for your support !能不能玩耍了

2016.09.23 16:47 (/p/61b73ac8662a/comments/4368486#comment-4368486)

回复

✎ 添加新回复



zero000 (/users/bda9686049bc)

8楼 2016.09.23 15:41 (/p/61b73ac8662a/comments/4367167#comment-4367167)

大神 我想问下，xcode7与xcode8是如何适配storyboard的

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @zero000 (/users/bda9686049bc) 我很少用storboard.没有专门研究过!
! 😊

2016.09.23 16:48 (/p/61b73ac8662a/comments/4368512#comment-4368512)

回复

✎ 添加新回复



GeeZher (/users/e3e0db5e1f54)

9楼 2016.09.23 16:49 (/p/61b73ac8662a/comments/4368531#comment-4368531)

#available(iOS 10.0, *)
为什么不用这种方式呢

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @GeeZher (/users/e3e0db5e1f54) 你是指__IOS_AVAILABLE(10.0);吗?
用这种方式做什么内容 😊

2016.09.23 18:21 (/p/61b73ac8662a/comments/4370397#comment-4370397)

回复

✎ 添加新回复



判若两人、 (/users/16ae66cdf6a0)

10楼 2016.09.25 15:25 (/p/61b73ac8662a/comments/4406495#comment-4406495)

通知内容都是我辛辛苦苦整理的，你就这么复制粘贴过来好么？

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @判若两人、 (/users/16ae66cdf6a0) 谢谢通知内容，只是希望大家可以共同学习。辛苦你了 😊

2016.09.26 15:21 (/p/61b73ac8662a/comments/4428291#comment-4428291)

回复

✎ 添加新回复

Regret_V (/users/b51fb333537e)



11 楼 · 2016.09.30 10:05 (/p/61b73ac8662a/comments/4506309#comment-4506309)

(/users/b51fb333537e)
总结的很棒!👍

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @Regret_V (/users/b51fb333537e) 谢谢支持😊

2016.09.30 12:09 (/p/61b73ac8662a/comments/4508415#comment-4508415)

回复

✎ 添加新回复



HHJZXL (/users/294a717ea284)

12 楼 · 2016.10.09 10:47 (/p/61b73ac8662a/comments/4690247#comment-4690247)
(/users/294a717ea284)

您好,想问你一个问题,就是我的程序在iOS10的真机运行之后,第一次安装时会出现系统的提示是否允许通知的提示框,点击“不允许”或“允许”之后线程直接死锁。死锁的原因是: 0x101e65c54 <+764>: brk #0x1 帮忙看一下这个是怎么解决的!谢谢了

♡ 喜欢(0)

回复

ouis (/users/026312b53954): @HHJZXL (/users/294a717ea284) 是不是用了腾讯bugly旧版本的,如果是的话,更新到最新的版本就好了

2016.10.12 16:07 (/p/61b73ac8662a/comments/4775457#comment-4775457)

回复

✎ 添加新回复



JaneYang1205 (/users/e098dbd985ab)

13 楼 · 2016.10.10 11:03 (/p/61b73ac8662a/comments/4715159#comment-4715159)
(/users/e098dbd985ab)

感谢分享

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @JaneYang1205 (/users/e098dbd985ab) 谢谢的认可😊

2016.10.10 13:54 (/p/61b73ac8662a/comments/4719934#comment-4719934)

回复

✎ 添加新回复



风逝留香 (/users/04ff13fdbfb3)

14楼 - 2016-10-10 11:23 (/p/61b73ac8662a/comments/4715745#comment-4715745)

你好。有个问题请教你一下。我在iOS10 上遇到了一个关于textfield密文输入的问题。获取焦点时的密文大小和失去焦点时密文text大小不一致。。但是监听他的font是一样的。请问又遇到过类似的问题吗？用了很多方法想去修正他。没有作用，也只在iOS10 上会出现

♡ 喜欢(0)

回复

Dely (/users/e95705fe04d9): @风逝留香 (/users/04ff13fdbfb3) 这个问题没有遇到过，我看看 😊

2016.10.10 13:54 (/p/61b73ac8662a/comments/4719920#comment-4719920)

回复

✎ 添加新回复



hhgvg (/users/ab910837438d)

15楼 - 2016-10-11 11:48 (/p/61b73ac8662a/comments/4743601#comment-4743601)

大神 我的web view加载本地html文件加载不出来

♡ 喜欢(0)

回复



十一岁的加重 (/users/1856923ecb16)

16楼 - 2016-10-11 17:57 (/p/61b73ac8662a/comments/4752156#comment-4752156)

Xcode8取消了三方插件的功能，好多教程破解可以继续使用，但是可能app上线可能会被拒。

实在搞不懂，Xcode用个插件，这个插件只是在你的源代码里，加入一些注释，而这些注释在打包的时候肯定会被过滤掉，怎么可能会因为用了这个插件而导致上架被拒呢

♡ 喜欢(0)

回复

加载更多 ↓ (/notes/5894755/comments?max_id=4775457&order=asc&page=2)

登录后发表评论 (/sign_in)

被以下专题收入，发现更多相似内容：



程序员 (/collection/NEt52a)

如果你是程序员，或者有一颗喜欢写程序的心，喜欢分享技术干货、项目经验、程序日常故事等，欢迎投稿《程序员》专题。专题主编：小...

24781篇文章 (/collection/NEt52a) · 185851人关注

+ 添加关注 (/sign_in)



iOS 开发 (/collection/2ffaa203eb6a)

8228篇文章 (/collection/2ffaa203eb6a) · 7094人关注 (/collection/2ffaa203eb6a)

+ 添加关注 (/sign_in)



iOS 大神班 (/collection/5aac963ca52d)

收集进阶知识、编程干货，每天会更新四篇。对于接收的文章，伯乐在线 (@iOS大神) 会联系您进行转载推广。如果您的投稿（私信投稿）...

863篇文章 (/collection/5aac963ca52d) · 2673人关注

+ 添加关注 (/sign_in)

A
➡
(/sign_in)