



[首页](#) [存储动态](#) [云存储](#) [存储技术](#) [网盘](#) [服务器](#) [资料下载](#) [磁盘阵列](#) [磁带存储](#) [存储招标信息](#) [存储论坛](#)

[首页](#) → [存储技术](#)

阅读新闻

背景: □□□□□□□□

iOS 开发 : 14个技巧助你适配 iOS10

[日期: 2016-09-19]

来源: 开源中国 作者:

[字体: 大 中 小]

1.Notification(通知)

自从 Notification 被引入之后, 苹果就不断的更新优化, 但这些更新优化只是小打小闹, 直至现在iOS 10开始真正的进行大改重构, 这让开发者也体会到 UserNotifications 的易用, 功能也变得非常强大。

• iOS 9 以前的通知

- 1.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。
- 2.应用在运行时和非运行时捕获通知的路径还不一致。
- 3.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。
- 4.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

- 1.所有相关通知被统一到了 UserNotifications.framework 框架中。
- 2.增加了撤销、更新、中途还可以修改通知的内容。
- 3.通知不再是简单的文本了, 可以加入视频、图片, 自定义通知的展示等等。
- 4.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。
- 5.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)。

2.ATS的问题

iOS 9中默认非HTTPS的网络是被禁止的, 当然我们也可以把 NSAllowsArbitraryLoads 设置为 YES 禁用 ATS 。不过iOS 10从2017年1月1日起苹果不允许我们通过这个方法跳过 ATS , 也就是说强制我们用 HTTPS , 如果不这样的话提交App可能会被拒绝。但是我们可以

Digg排行

- 5 5分钟了解VMware vSAN的分布式
- 3 2015年存储TOP5认证, 高薪必备
- 2 从 MVC 到前后端分离
- 2 图文详解 Andorid 中用 Shape 定
- 2 Android 三种播放视频的方式
- 2 1克DNA干掉4T硬盘 存储新革命?
- 2 各大厂vSphere Metro Storage
- 2 vSphere 6.0 Virtual Volumes架
- 2 几乎所有的硬盘都有了后门,存储
- 2 此尤物一出, EMC Isilon和华为

热门评论

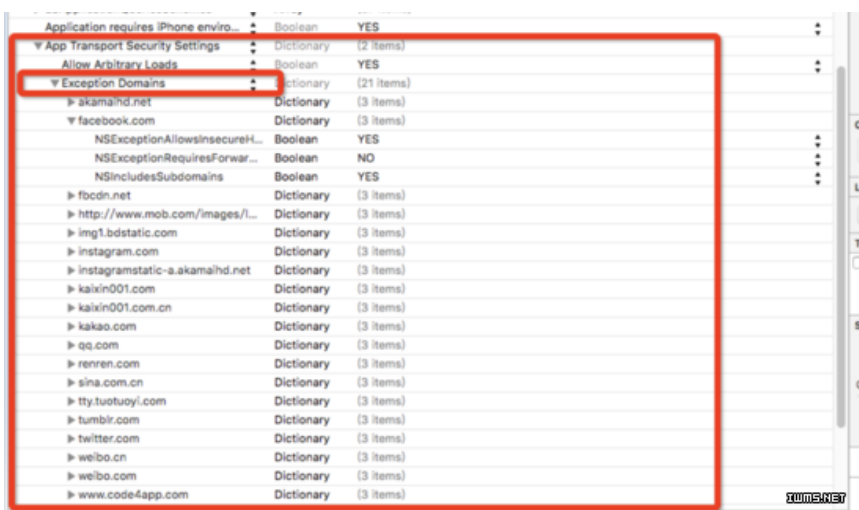
以通过 `NSExceptionDomains` 来针对特定的域名开放 HTTP 可以容易通过审核。

`NSExceptionDomains`方式 设置域。可以简单理解成，把不支持https协议的接口设置成http的接口。

具体方法：

- 1)、在项目的info.plist中添加一个Key: App Transport Security Settings, 类型为字典类型。
- 2)、然后给它添加一个Exception Domains, 类型为字典类型;
- 3)、把需要的支持的域添加给Exception Domains。其中域作为Key, 类型为字典类型。
- 4)、每个域下面需要设置3个属性: `NSIncludesSubdomains`、`NSExceptionRequiresForwardSecrecy`、`NSExceptionAllowsInsecureHTTPLoads`。

如图：



细节提示:在iOS9以后的系统中如果使用到网络图片，也要注意网络图片是否是HTTP的哦，如果是，也要把图片的域设置哦！

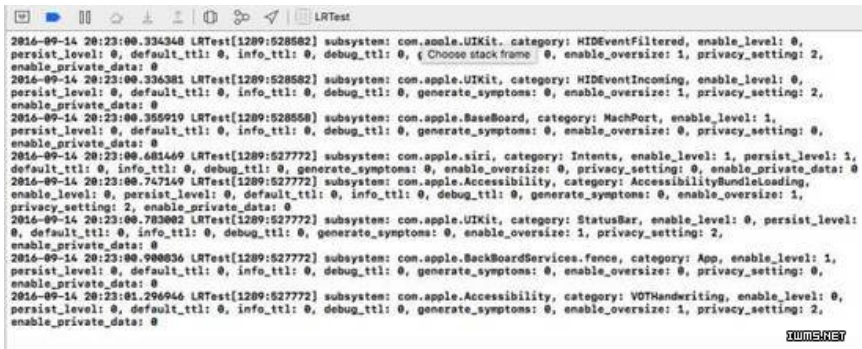
3.iOS 10 隐私权限设置

iOS 10 开始对 隐私权限 更加严格，如果你不设置就会直接崩溃，现在很多遇到崩溃问题了，一般解决办法都是在 `info.plist` 文件添加对应的 Key - Value 就可以了。

Key	Type	Value
▼ Information Property List	Dictionary (25 items)	对应需要访问的属性
Privacy - Photo Library Usage Description	String	访问相册
Privacy - Camera Usage Description	String	访问相机
Privacy - Location Usage Description	String	访问位置
Privacy - Location Always Usage Description	String	始终访问位置
Privacy - Location When In Use Usage Description	String	使用期间访问位置
Privacy - Bluetooth Peripheral Usage Description	String	访问蓝牙
Privacy - Calendars Usage Description	String	访问日历
Privacy - Reminders Usage Description	String	访问提醒事项
Privacy - Microphone Usage Description	String	访问麦克风
Privacy - Media Library Usage Description	String	访问媒体资料库
Privacy - Motion Usage Description	String	访问运动与健康

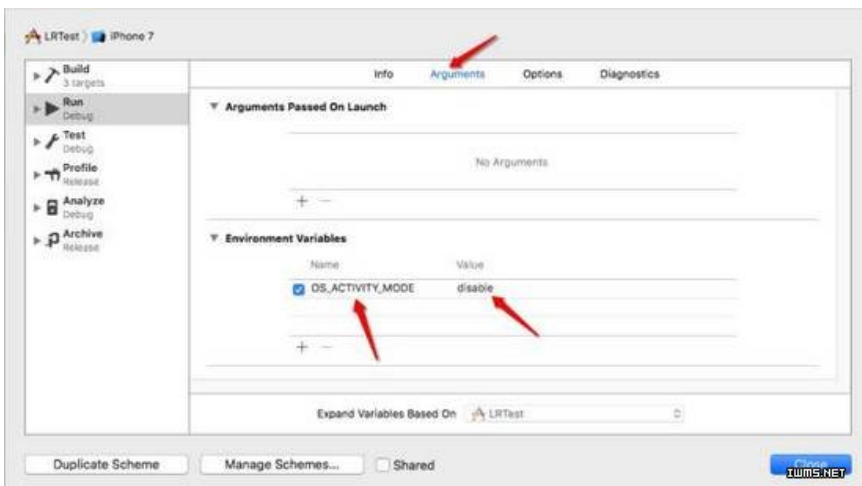
以上 Value 值，圈出的红线部分的文字是展示给用户看的，必须添加。

4.Xcode 8 运行一堆没用的logs解决办法



上图我们看到，自己新建的一个工程啥也没干就打印一堆烂七八糟的东西，我觉得这个应该是 Xcode 8 的问题，

具体也没细研究，解决办法是设置 OS_ACTIVITY_MODE : disable 如下图：



5.iOS 10 UIStatusBar方法过期:

```
// Setting the statusBarStyle does nothing if your application is using the
// default UIViewController-based status bar system.
@property(readwrite, nonatomic) UIStatusBarStyle statusBarStyle NS_DEPRECATED_IOS(
2_0, 9_0, "Use -[UIViewController preferredStatusBarStyle]")
__TVOS_PROHIBITED;
- (void)setStatusBarStyle:(UIStatusBarStyle)statusBarStyle animated:(BOOL)
animated NS_DEPRECATED_IOS(2_0, 9_0, "Use -[UIViewController
preferredStatusBarStyle]") __TVOS_PROHIBITED;

// Setting statusBarHidden does nothing if your application is using the default
// UIViewController-based status bar system.
@property(readwrite, nonatomic, getter=isStatusBarHidden) BOOL statusBarHidden
NS_DEPRECATED_IOS(2_0, 9_0, "Use -[UIViewController prefersStatusBarHidden]")
__TVOS_PROHIBITED;
- (void)setStatusBarHidden:(BOOL)hidden withAnimation:(UIStatusBarAnimation)
animation NS_DEPRECATED_IOS(3_2, 9_0, "Use -[UIViewController
prefersStatusBarHidden]") __TVOS_PROHIBITED;
- (BOOL)setKeepAliveTimeout:(NSTimeInterval)timeout handler:(void(^ __nullable)
(void))keepAliveHandler NS_DEPRECATED_IOS(4_0, 9_0, "Please use
UIRemoteNotificationTypeVoIP remote notifications for VoIP applications")
__TVOS_PROHIBITED;
- (void)clearKeepAliveTimeout NS_DEPRECATED_IOS(4_0, 9_0, "Please use
UIRemoteNotificationTypeVoIP remote notifications for VoIP applications")
__TVOS_PROHIBITED;
```

在我们开发中有可能用到 UIStatusBar 一些属性，在iOS 10 中这些方法已经过期了，如果你的项目中有用的话就得需要适配。

上面的图片也能发现，如果在iOS 10中你需要使用 preferredStatusBar 比如这样：

```
//iOS 10 - (UIStatusBarStyle)preferredStatusBarStyle { return
UIStatusBarStyleDefault;
}
```


6.iOS 10 UICollectionView 性能优化

随着开发者对 UICollectionView 的信赖,项目中用的地方也比较多,但是还是存在一些问题,比如有时会卡顿、加载慢等。所以iOS 10 对 UICollectionView 进一步的优化。

- UICollectionView cell pre-fetching预加载机制
- UICollectionView and UITableView prefetchDataSource 新增的API
- 针对self-sizing cells 的改进
- Interactive reordering

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:

- 1.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;
- 2.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;
- 3.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;
- 4.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:

- 1.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;
- 2.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;
- 3.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;
- 4.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:(NSArray<NSIndexPath *> *)indexPaths NS_AVAILABLE_IOS(10_0);

- (void)collectionView:(UICollectionView *)collectionView cancelPrefetchingForItemsAtIndexPaths:(NSArray<NSIndexPath *> *)indexPaths NS_AVAILABLE_IOS(10_0);
```

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.

7.iOS 10 UIColor 新增方法

以下是官方文档的说明:

Most graphics frameworks throughout the system, including Core Graphics, Core Image, Metal, and AVFoundation, have substantially improved support for extended-range pixel formats and wide-gamut color spaces. By extending this behavior throughout the entire graphics stack, it is easier than ever to support devices with a wide color display. In addition, UIKit standardizes on working in a new extended sRGB color space, making it easy to mix sRGB colors with colors in other, wider color gamuts without a significant performance penalty.

Here are some best practices to adopt as you start working with Wide Color.

- In iOS 10, the UIColor class uses the extended sRGB color space and its initializers no longer clamp raw component values to between 0.0 and 1.0. If your app relies on UIKit to clamp component values (whether you're creating a color or asking a color for its component values), you need to change your app's behavior when you link against iOS 10.
- When performing custom drawing in a UIView on an iPad Pro (9.7 inch), the underlying drawing environment is configured with an extended sRGB color space.
- If your app renders custom image objects, use the new UIGraphicsImageRenderer class to control whether the destination bitmap is created using an extended-range or standard-range format.
- If you are performing your own image processing on wide-gamut devices using a lower level API, such as Core Graphics or Metal, you should use an extended range color space and a pixel format that supports 16-bit floating-point component values. When clamping of color values is necessary, you should do so explicitly.
- Core Graphics, Core Image, and Metal Performance Shaders provide new options for easily converting colors and images between color spaces.

因为之前我们都是用 RGB 来设置颜色,反正用起来也不是特别多样化,这次新增的方法应该就是一个弥补吧。所以在iOS 10 苹果官方建议我们使用 sRGB ,因为它性能更好,色彩更

丰富。如果你自己为 `UIColor` 写了一套分类的话也可尝试替换为 `sRGB`，`UIColor` 类中新增了两个 `Api` 如下：

```
+ (UIColor *)colorWithDisplayP3Red:(CGFloat)displayP3Red green:(CGFloat)green
blue:(CGFloat)blue alpha:(CGFloat)alpha NS_AVAILABLE_IOS(10_0);

- (UIColor *)initWithDisplayP3Red:(CGFloat)displayP3Red green:(CGFloat)green
blue:(CGFloat)blue alpha:(CGFloat)alpha NS_AVAILABLE_IOS(10_0);
```

8.iOS 10 UITextContentType

```
// The textContentType property is to provide the keyboard with extra information
about the semantic intent of the text document.@property(nonatomic,copy)
UITextContentType textContentType NS_AVAILABLE_IOS(10_0); // default is nil
```

在iOS 10 `UITextField` 添加了 `textContentType` 枚举，指示文本输入区域所期望的语义意义。

使用此属性可以给键盘和系统信息，关于用户输入的内容的预期的语义意义。例如，您可以指定一个文本字段，用户填写收到一封电子邮件确认 `uitextcontenttypeemailaddress`。当您提供有关您期望用户在文本输入区域中输入的内容的信息时，系统可以在某些情况下自动选择适当的键盘，并提高键盘修正和主动与其他文本输入机会的整合。

9.iOS 10 字体随着手机系统字体而改变

当我们手机系统字体改变了之后，那我们 `App` 的 `label` 也会跟着一起变化，这需要我们写很多代码来进一步处理才能实现，但是iOS 10 提供了这样的属性 `adjustsFontForContentSizeCategory` 来设置。因为没有真机，具体实际操作还没去实现，如果理解错误帮忙指正。

```
UILabel *myLabel = [UILabel new]; /*
    UIFont 的preferredFontForTextStyle: 意思是指定一个样式，并让字体大小符合用户设置的
    字体大小。
    */
    myLabel.font =[UIFont preferredFontForTextStyle: UIFontTextStyleHeadline]
; /*
    Indicates whether the corresponding element should automatically update its
    font when the device's UIContentSizeCategory is changed.
    For this property to take effect, the element's font must be a font vended u
    sing +preferredFontForTextStyle: or +preferredFontForTextStyle:compatibleWith
    TraitCollection: with a valid UIFontTextStyle.
    */
    //是否更新字体的变化
    myLabel.adjustsFontForContentSizeCategory = YES;
```


10.iOS 10 UIScrollView新增refreshControl

UIScreen.h

```
Added UIScreen.focusedItem
Added UIScreen.mainScreen
Added UIScreen.screens
```

UIScrollView.h

```
Added UIScrollView.refreshControl
```

UISelectionFeedbackGenerator.h (Added)

```
Added UISelectionFeedbackGenerator
Added -[UISelectionFeedbackGenerator selectionChanged]
```

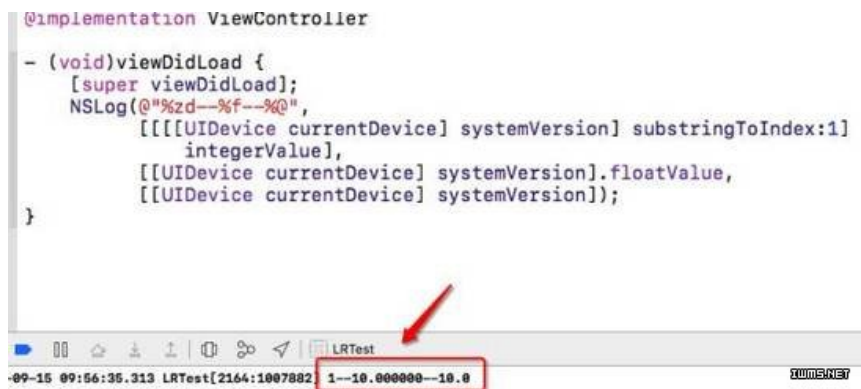
TUMS.NET

iOS 10 以后只要是继承 UIScrollView 那么就支持刷新功能:

```
@property (nonatomic, strong, nullable) UIRefreshControl *refreshControl NS_AVAILABLE_IOS(10_0) __TVOS_PROHIBITED;
```

11.iOS 10 判断系统版本正确姿势

判断系统版本是我们经常用到的, 尤其是现在大家都有可能需要适配iOS 10, 那么问题就出现了, 如下图:



我们得到了答案是:

```
//值为 1 [[[UIDevice currentDevice] systemVersion] substringToIndex:1] integerValue]
```

```
//值为10.000000 [[[UIDevice currentDevice] systemVersion].floatValue,
```

```
//值为10.0 [[[UIDevice currentDevice] systemVersion]
```

所以说判断系统方法最好还是用后面的两种方法, 哦~我忘记说了 [[UIDevice currentDevice] systemVersion].floatValue 这个方法也是不靠谱的, 好像在 8.3 版本输出的值是 8.2, 记不清楚了反正是不靠谱的, 所以建议大家用 [[UIDevice currentDevice] systemVersion] 这个方法!

Swift判断如下:

```
if #available(iOS 10.0, *) {
    // iOS 10.0
    print("iOS 10.0");
} else { }
```

12.Xcode 8 插件不能用的问题

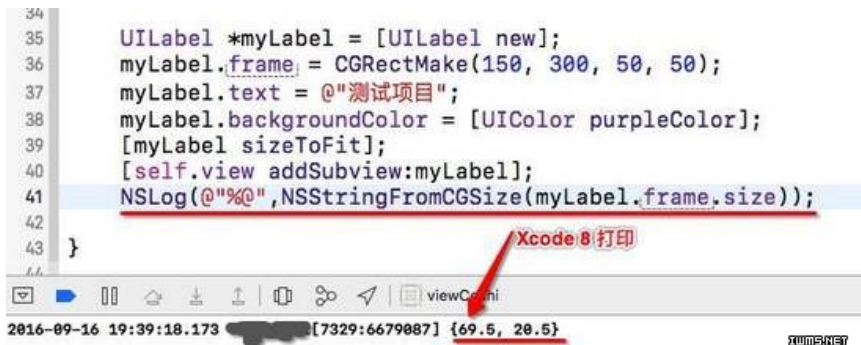
大家都升级了 Xcode 8，但是对于插件依赖的开发者们，一边哭着一边去网上寻找解决办法。那么下面是解决办法：

让你的 Xcode8 继续使用插件(http://vongloo.me/2016/09/10/Make-Your-Xcode8-Great-Again/?utm_source=tuicool&utm_medium=referral)

但是看到文章最后的解释，我们知道如果用插件的话，可能安全上会有问题、并且提交审核会被拒绝，所以建议大家还是不要用了，解决办法总是有的，比如在 Xcode 中添加注释的代码块也是很方便的。

13.iOS 10开始项目中有的文字显示不全问题

我用 Xcode 8 和 Xcode 7.3 分别测试了下，如下图：

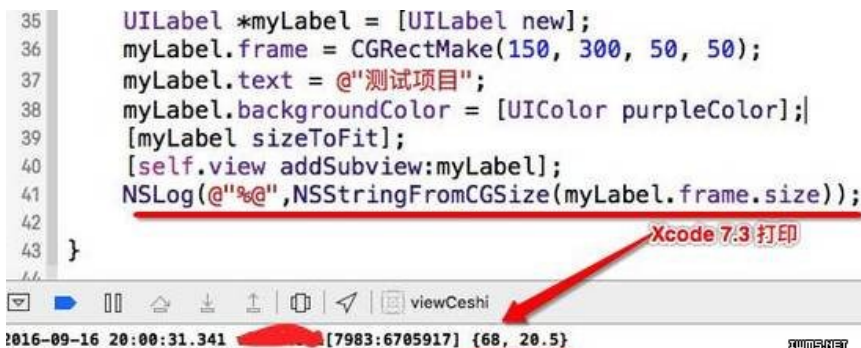


```
34
35 UILabel *myLabel = [UILabel new];
36 myLabel.frame = CGRectMake(150, 300, 50, 50);
37 myLabel.text = @"测试项目";
38 myLabel.backgroundColor = [UIColor purpleColor];
39 [myLabel sizeToFit];
40 [self.view addSubview:myLabel];
41 NSLog(@"%@", NSStringFromCGSize(myLabel.frame.size));
42
43 }
```

Xcode 8 打印

2016-09-16 19:39:18.173 [7329:6679087] {69.5, 20.5}

Xcode 8



```
35 UILabel *myLabel = [UILabel new];
36 myLabel.frame = CGRectMake(150, 300, 50, 50);
37 myLabel.text = @"测试项目";
38 myLabel.backgroundColor = [UIColor purpleColor];
39 [myLabel sizeToFit];
40 [self.view addSubview:myLabel];
41 NSLog(@"%@", NSStringFromCGSize(myLabel.frame.size));
42
43 }
```

Xcode 7.3 打印

2016-09-16 20:00:31.341 [7983:6705917] {68, 20.5}

Xcode7

创建一个
Label

然后让它自适应大小，字体大小都是

17

最后输出的宽度是不一样的，我们再看一下，
下面的数据就知道为什么升级

iOS 10

之后

App

中有的文字显示不全了：



同声译

跟外国人交流有困难？您需要带上同声翻译官！直接对着手机说话，就能翻译并能说出来，中译英，英译中。您能听懂老外的话了，老外能听懂你的话。接待外宾的神器！出国旅行的神器！

Xcode 8打印	Xcode 7.3打印
1个文字宽度：17.5	1个文字宽度：17
2个文字宽度：35	2个文字宽度：34
3个文字宽度：52	3个文字宽度：51
4个文字宽度：69.5	4个文字宽度：68
5个文字宽度：87	5个文字宽度：85
6个文字宽度：104	6个文字宽度：102
7个文字宽度：121.5	7个文字宽度：119
8个文字宽度：139	8个文字宽度：136
9个文字宽度：156	9个文字宽度：153
10个文字宽度：173.5	10个文字宽度：170

英文字母会不会也有这种问题，我又通过测试，后来发现英文字母没有问题，只有汉字有问题。目前只有一个一个修改控件解决这个问题，暂时没有其他好办法来解决。

14.Xcode 8使用Xib awakeFromNib的警告问题

在 Xcode 8 之前我们使用 Xib 初始化 - (void)awakeFromNib {} 都是这么写也没什么问题，但是在 Xcode 8 会有如下警告：

```
- (void)awakeFromNib {  
    //[super awakeFromNib];  
}
```

Method possibly missing a [super awakeFromNib] call

官方解释：
You must **call** the super implementation **of** awakeFromNib **to** give parent classes the opportunity **to** perform **any** additional initialization they require. Although the **default** implementation **of** this method does nothing, many UIKit classes provide non-empty implementations.
You may **call** the super implementation **at any** point during your own awakeFromNib method.

你必须调用父类实现awakeFromNib来给父类来执行它们需要的任何额外的初始化的机会。虽然这种方法的默认实现不做任何事情，许多UIKit类提供非空的实现。你可以调用自己的awakeFromNib方法中的任何时候超级实现。



随时随地告诉你最精确的PM2.5值！走南闯北，随时查看，PM2.5值随你而变！不是城市的平均值，是用离你最近的数个PM2.5传感器，在云计算平台精确计算出来的实时值。还可接入酷炫的“我的PM2.5”，配套智能硬件，实时监测室内空气质量。

分享

赶快成为第一个分享的人吧

收藏

推荐

打印

| 录入: Cstor | 阅读: 201 次

从 MVC 到前后端分离

Python子进程

科技头条



扫二维码下载APP

(目前仅支持Android)

看新闻，发广告：每次评论均显示您的广告词和链接！

科技头条——科技爱好者的情报站，是一款挖掘互联网科技

类新闻、微信公众账号热点文章的阅读客户端。酷炫科技，远见

未来。精选微信朋友圈100000+的热点文章、知名微信公众

账号，每日智能推送您关心的内容，帮助您更加高效地获取科

技信息、追踪科技热点。



中国存储资讯



云创大数据



中国智慧城市



大数据资讯



中国_物联网



中国云计算论坛



中国PM25资讯



专注云计算



笑看云存储

一键关注

注册微博

相关新闻

iOS

- iOS应用UI线程卡顿监控 (今 09:13)
- 用Python为iOS10生成图标和截屏 (09月22日)
- iOS 中的 block 是如何持有对象的 (08月10日)
- iOS开发之iOS10推送必看 (09月28日)
- iOS函数响应式编程以及ReactiveCoc (08月12日)

本文评论

查看全部评论 (0)

表情:

姓名: ☒ 匿名 字数 0

点评:

☒ 同意评论声明

发表

评论声明

- 尊重网上道德，遵守中华人民共和国的各项有关法律法规
- 承担一切因您的行为而直接或间接导致的民事或刑事责任
- 本站管理人员有权保留或删除其管辖留言中的任意内容
- 本站有权在网站内转载或引用您的评论
- 参与本评论即表明您已经阅读并接受上述条款

版权所有:© 2013-2016 南京云创大数据科技股份有限公司（股票代码：835305），保留一切权利。苏ICP备11060547号-4

本站所转载、链接、引用及翻译的内容等，其版权属于各自合法的拥有者。

本站创建、管理、商务合作: zhujiguangjie#163.com(用@代替#)



Powered by iwms 6.0