



作者 seedante (/users/7ba5d9065301) 2015.06.24 13:23\*

写了103437字, 被1712人关注, 获得了691个喜欢  
(/users/7ba5d9065301)

+ 添加关注 (/sign\_in)

# Photos 框架实践以及坑

字数3392 阅读15035 评论20 喜欢50

## 导读

对于 Photos 框架的介绍, 推荐观看 objccn.io 的文章 (<http://objccn.io/issue-21-4/>)。写得真好, 我写得文章水准还差得老远啊。本文总结了近期使用 Photos 框架编写一个相册的经验, 目前还有很大一部分的框架内容没有涉及到, 后续会更新内容。

## 获取资源

照片库中有两种资源可供获取: PHAsset 和 PHCollection, 前者代表图像或视频对象, 后者是前者的集合或自身类型的集合。PHCollection 是个基类, 有 PHAssetCollection 和 PHCollectionList 两个子类, 分别代表 Photos 里的相册和文件夹。以往使用 Photos 时, 并没有注意到可以建立文件夹, 似乎是从 Photos 框架才支持这个功能, 而 PHCollectionList 里可嵌套 PHAssetCollection 和自身类型, 还支持多重嵌套。获取 PHAsset 以及 PHAssetCollection 的过程类似于 Core Data, 如下所示, 只能通过类方法来返回 PHFetchResult, 遍历返回的结果来获取需要的资源。

### Fetching Assets

```
M + fetchAssetsInAssetCollection:options:
M + fetchAssetsWithMediaType:options:
M + fetchAssetsWithLocalIdentifiers:options:
M + fetchKeyAssetsInAssetCollection:options:
M + fetchAssetsWithOptions:
M + fetchAssetsWithBurstIdentifier:options:
M + fetchAssetsWithALAssetURLs:options:
```

*PHAsset Fetch Method*

```
Fetching Asset Collections
[M] + fetchAssetCollectionsWithLocalIdentifiers:options:
[M] + fetchAssetCollectionsWithType:subtype:options:
[M] + fetchAssetCollectionsContainingAsset:withType:options:
[M] + fetchAssetCollectionsWithALAssetGroupURLs:options:
[M] + fetchMomentsInMomentList:options:
[M] + fetchMomentsWithOptions:
```

*PHAssetCollection Fetch Method*

注意，PHAsset、PHAssetCollection 和 PHCollectionList 都是轻量级的不可变对象，使用这些类时并没有将其代表的图像或视频或是集合载入内存中，要使用其代表的图像或视频，需要通过 PHImageManager 类来请求。

## 请求图像(这里有巨坑)

关于 PHImageManager ([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHImageManager\\_Class/index.html#//apple\\_ref/doc/uid/TP40014399](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHImageManager_Class/index.html#//apple_ref/doc/uid/TP40014399))类，NSHipster 有篇总结文章 (<http://nshipster.com/phimagemanager/>)不错。

```
- requestImageForAsset:targetSize:contentMode:options:resultHandler:
```

你不应该生成该类的实例，而应该使用该类的提供的单例对象。该方法提供指定的尺寸的图像，与 ALAssetsLibrary 库相比，没有了方便的缩略图提供。不过要吐槽的是，ALAssetsLibrary 库提供的缩略图往往尺寸太小并且质量很低，用在 TableView 上还可以。

需要注意的是，该方法在默认情况下是异步执行的，而且 Photos 库可能会多次执行 resultHandler 块，因为对于指定的尺寸，Photos 可能会先提供低质量的图像以供临时显示，随后会将指定尺寸的图像返回。如果指定尺寸的高质量的图像有缓存，那么直接提供高质量的图像。而这些行为，可以通过 options 参数来定制。

PHImageRequestOptions

([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHImageRequestOptions\\_Class/index.html#//apple\\_ref/swift/cl/c:objc\(cs\)PHImageRequestOptions](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHImageRequestOptions_Class/index.html#//apple_ref/swift/cl/c:objc(cs)PHImageRequestOptions))类用于定制请求。**这里有巨坑**。上面的方法返回指定尺寸的图像，如果你仅仅指定必要的参数而没有对 options 进行配置的话，返回的图像尺寸将会是原始图像的尺寸。或

者，你指定的尺寸很小，这时候会按照你的要求来返回接近该尺寸的图像。在我的 iPad mini 一代上，对于自拍的图像，指定尺寸不超过(257, 257)的话，返回的图像尺寸和你预期的一样，其他情况下都是原始尺寸。 `PHImageRequestOptions` ([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHImageRequestOptions\\_Class/index.html#//apple\\_ref/swift/cl/c:objc\(cs\)PHImageRequestOptions](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHImageRequestOptions_Class/index.html#//apple_ref/swift/cl/c:objc(cs)PHImageRequestOptions))有以下几个重要的属性：

`synchronous`：指定请求是否同步执行。

`resizeMode`：对请求的图像怎样缩放。有三种选择：None，不缩放；Fast，尽快地提供接近或稍微大于要求的尺寸。

`deliveryMode`：图像质量。有三种值：Opportunistic，在速度与质量中均衡；HighQualityFormat，不管在什么情况下都提供最高质量。这个属性只有在 `synchronous` 为 true 时有效。

`normalizedCropRect`：用于对原始尺寸的图像进行裁剪，基于比例坐标。只在 `resizeMode` 为 Exact 时有效。

`resizeMode` 默认是 None，这也造成了返回图像尺寸与要求尺寸不符。这点需要注意。要返回一个指定尺寸的图像需要避免两层陷阱：一定要指定 `options` 参数，`resizeMode` 不能为 None。

除了必有请求图像或是视频的功能外， `PHImageManager` 添加了两大功能：

1. 缓存图像，由其子类 `PHCachingImageManager` ([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHCachingImageManager\\_Class/index.html#//apple\\_ref/occ/cl/PHCachingImageManager](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHCachingImageManager_Class/index.html#//apple_ref/occ/cl/PHCachingImageManager)) 实现，缓存效率和空间管理能满足大部分场景的需求；
2. 裁剪图像，这个功能很久以前就有强烈的需求。六年前 StackOverflow 上 Cropping a UIImage (<http://stackoverflow.com/questions/158914/cropping-a-uiimage>) 这个问题就被提出来了，方法也五花八门，然而这些方法可能会有各种小问题。官方的方法能让你避免这些小问题。使用方法可以参考 NSHipster 的总结文章里用人脸识别获取头像的例子。

## localIdentifier vs URL

Photos 框架推出时，和原来的照片库 `AssetsLibrary` 框架之间还有些交互， `PHAsset` 类的 `+ fetchAssetsWithALAssetURLs:options:` 和 `PHAssetCollection` 类的 `+ fetchAssetCollectionsWithALAssetGroupURLs:options:` 可以利用原来的 `AssetsLibrary` 提供的 URL 进行转化，而在 iOS 9 中，原来的照片框架 `AssetsLibrary` 已经被废弃了，如今这两个方法也没有用处了。当初我还找过如何从 Photos 框架到 `AssetsLibrary` 框架的方法，理所当然地白费功夫，官方要淡化照片库中 URL 的概念，改之使用一个标志符来唯一代表一个资源。Photos 框架中的根类 `PHObject` 只有一个公开接口 `localIdentifier`，

AssetsLibrary 框架中无论是 Asset 还是 AssetGroup 的 URL 也是唯一标志符，而且同时还是动态变化的，每次启动应用后获取的 URL 和上一次是不一样的，而 AssetGroup 有一个 PersistentID 与 PHObject 的 localIdentifier 类似，但获取比较麻烦。

localIdentifier 属性带来的最大好处是 PHObject 类实现了 NSCopying 协议，可以直接使用 localIdentifier 属性对 PHObject 及其子类对象进行对比是否同一个对象。

## 获取指定类型相册

这是最基本的一个用途，但是每次隔了几天就忘了具体的类型。

通过 PHAssetCollection 的以下方法来获取指定的相册：

```
func fetchAssetCollectionsWithType(_ type: PHAssetCollectionType, subtype subtype: PHAssetCollectionSubtype) throws -> [PHAssetCollection]
```

这个方法需要至少指定两个参数：

```
enum PHAssetCollectionType : Int {
    case Album //从 iTunes 同步来的相册，以及用户在 Photos 中自己建立的相册
    case SmartAlbum //经由相机得来的相册
    case Moment //Photos 为我们自动生成的时间分组的相册
}

enum PHAssetCollectionSubtype : Int {
    case AlbumRegular //用户在 Photos 中创建的相册，也就是我所谓的逻辑相册
    case AlbumSyncedEvent //使用 iTunes 从 Photos 照片库或者 iPhoto 照片库同步过来的事件。系统相册。
    case AlbumSyncedFaces //使用 iTunes 从 Photos 照片库或者 iPhoto 照片库同步的人物相册。系统相册。
    case AlbumSyncedAlbum //做了 AlbumSyncedEvent 应该做的事
    case AlbumImported //从相机或是外部存储导入的相册，完全没有这方面的使用经验，没法验证。
    case AlbumMyPhotoStream //用户的 iCloud 照片流
    case AlbumCloudShared //用户使用 iCloud 共享的相册
    case SmartAlbumGeneric //文档解释为非特殊类型的相册，主要包括从 iPhoto 同步过来的相册。由于 iPhoto 已经废弃，所以这个类型已经很少见了。
    case SmartAlbumPanoramas //相机拍摄的全景照片
    case SmartAlbumVideos //相机拍摄的视频
    case SmartAlbumFavorites //收藏文件夹
    case SmartAlbumTimelapses //延时视频文件夹，同时也会出现在视频文件夹中
    case SmartAlbumAllHidden //包含隐藏照片或视频的文件夹
    case SmartAlbumRecentlyAdded //相机近期拍摄的照片或视频
    case SmartAlbumBursts //连拍模式拍摄的照片，在 iPad mini 上按住快门不放就可以了，但是照片库中只保留一张照片
    case SmartAlbumSlomoVideos //Slomo 是 slow motion 的缩写，高速摄影慢动作解析，在该模式下拍摄的视频
    case SmartAlbumUserLibrary //这个命名最神奇了，就是相机相册，所有相机拍摄的照片或视频都会出现在这里
    case Any //包含所有类型
}
```

有些参数的命名十分令人困惑，我每次看了都晕菜。新的 Photos Kit 框架是在 iOS 8 中推出的，主类型分为三种类型：Album，SmartAlbum 以及 Moment。然而，对于前两者的分类我是比较困惑的。Mac 上支持智能文件夹，就是可以不管文件的物理位置而将一系列文件集合起来建立一个文件夹，可以说是**物理相册和逻辑相册**。而在 iOS 上，SmartAlbum 却给了相机衍生的相册，用户收集不同照片建立的逻辑相册被归类到 Album 类型的 AlbumRegular 下，十分反我的直觉。在文档中，SmartAlbum 是指内容会动态变化的相册，这样一来又有一个比较困惑的设计，PHAssetCollection 类有个属性 `estimatedAssetCount`，可以用来快速获取该相册中的照片和视频的数量，但是在 SmartAlbum 上该属性永远为 0，动态相册没能实现对数量的监测。

注意，获取指定类型的相册时，主类型和子类型要匹配，不要串台。如果不匹配，系统会按照 Any 子类型来处理。对于 Moment 类型，子类型使用 Any。

1. 获取用户自己建立的相册和文件夹（我称之为逻辑相册，非系统相册和从 iTunes 同步来的相册）有两种方法：

```
PHCollection.fetchTopLevelUserCollectionsWithOptions(nil)
PHAssetCollection.fetchAssetCollectionsWithType(.Album, subtype: .AlbumRegular, opti
```

在没有提供 PHOptions 的情况下，返回的 PHFetchResult 结果是按相册的建立时间排序的，最新的在前面。

2. 获取相机相册：

```
PHAssetCollection.fetchAssetCollectionsWithType(.SmartAlbum, subtype: .SmartAlbumUse
```

**另外：**PHAsset 的获取方式在 iOS 8.1 后发生了一些变化。以下的两个方法在 iOS 8.1 后不再包含从 iTunes 同步以及在 iCloud 中的照片和视频。要获取 iOS 设备上本地的所有照片和资源只能从 PHAssetCollection 入手了。

```
+ fetchAssetsWithMediaType:options:
+ fetchAssetsWithOptions:
```

## 添加、删除、编辑

对照片库进行操作，可参见官方文档 [Requesting Changes to the Photo Library](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHPhotoLibrary_Class/) ([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHPhotoLibrary\\_Class/](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHPhotoLibrary_Class/))，照片库中的资源都有对应的变更请求类：`PHAssetChangeRequest` ([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHAssetChangeRequest\\_Class/index.html#//apple\\_ref/occ/cl/PHAssetChangeRequest](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHAssetChangeRequest_Class/index.html#//apple_ref/occ/cl/PHAssetChangeRequest)), `PHAssetCollectionChangeRequest` ([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHAssetCollectionChangeRequest\\_Class/index.html#//apple\\_ref/occ/cl/PHAssetCollectionChangeRequest](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHAssetCollectionChangeRequest_Class/index.html#//apple_ref/occ/cl/PHAssetCollectionChangeRequest)) 和 `PHCollectionListChangeRequest` ([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHCollectionListChangeRequest\\_Class/index.html#//apple\\_ref/occ/cl/PHCollectionListChangeRequest](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHCollectionListChangeRequest_Class/index.html#//apple_ref/occ/cl/PHCollectionListChangeRequest)), 而这些操作的请求都要求在 `PHPhotoLibrary` 的 `performChanges(_ changeBlock: dispatch_block_t!, completionHandler completionHandler: ((Bool, NSError!) -> Void)!)` 中的 `changeBlock` 中执行。注意，这里只是发出请求并没有做出实质的更改，因此想要根据更改结果更新 UI 的话不要在 `completionHandler` 中进行，而应该在 `photoLibraryDidChange(changeInfo: PHChange!)` 中进行。三种变更请求中，删除和编辑操作都比较简单，而添加操作有需要注意的地方。

### 添加操作: placeholder 的用处

在相册中添加照片：

```
let createAssetRequest = PHAssetChangeRequest.creationRequestForAssetFromImage(image)
let assetPlaceholder = createAssetRequest.placeholderForCreatedAsset
let albumChangeRequest = PHAssetCollectionChangeRequest(forAssetCollection: album)
albumChangeRequest.addAssets([assetPlaceholder])
```

在文件夹中添加相册：

```
let fetchResult = PHCollection.fetchCollectionsInCollectionList(collectionList, options: nil)
let createSubAlbumRequest = PHAssetCollectionChangeRequest.creationRequestForAssetCollection
let albumPlaceholder = createSubAlbumRequest.placeholderForCreatedAssetCollection
let folderChangeRequest = PHCollectionListChangeRequest.init(forCollectionList: collectionList)
folderChangeRequest?.addChildCollections([albumPlaceholder])
```

在文件夹中添加子文件夹：

```
let fetchResult = PHCollection.fetchCollectionsInCollectionList(collectionList, opti
let createSubFolderRequest = PHCollectionListChangeRequest.creationRequestForCollect
let subfolderPlaceholder = createSubFolderRequest.placeholderForCreatedCollectionLis
let folderChangeRequest = PHCollectionListChangeRequest.init(forCollectionList: coll
folderChangeRequest?.addChildCollections([subfolderPlaceholder])
```

## 处理变更

对相册发出变更请求后，系统会通知用户是否允许，用户允许后才会发生实质上的变化，系统会发布通知。

首先，注册成为 PHPhotoLibrary 的观察者来接收变化通知：

➡ 登录 (/sign\_in) 👤 注册 (/sign\_up)

```
PHPhotoLibrary.shareLibrary().registerChangeObserver(self)
```

然后，实现 PHPhotoLibraryChangeObserver 协议的 photoLibraryDidChange(changeInfo: PHChange!)。官方有个很好的例子：Handling Changes: An Example ([https://developer.apple.com/library/ios/documentation/Photos/Reference/PHPhotoLibraryChangeObserver\\_Protocol/index.html#//apple\\_ref/occ/intf/PHPhotoLibraryChangeObserver](https://developer.apple.com/library/ios/documentation/Photos/Reference/PHPhotoLibraryChangeObserver_Protocol/index.html#//apple_ref/occ/intf/PHPhotoLibraryChangeObserver))，有以下几点需要注意：

1. 在 photoLibraryDidChange(changeInfo: PHChange!) 的实现里将所有处理放在主线程里处理；
2. 所有 PHPhotoLibrary 的观察者都会收到通知，不管观察者本身引用的内容是否发生变化，因此要根据观察者的情况来对通知进行过滤。从参数 PHChange 对象里能获得所有的变化，通过 changeDetailsForObject: 和 changeDetailsForFetchResult: 来获取细节。changeDetailsForObject: 获取的细节只是 PHObject 子类对象本身的信息变化，包括是否有成员被删除以及是否有图像或视频发生变化两种信息，有用信息实在有限，要处理成员变化需要依靠后者；对一个 PHFetchResult 对象使用 changeDetailsForFetchResult: 获取的细节中只包含该 PHFetchResult 对象变化的信息，可以利用这点来对通知进行过滤处理。
3. 通过 changeDetailsForFetchResult: 获取的 PHFetchResultChangeDetails 对象，包含了 FetchResult 的结果的所有变化情况以及 FetchResult 的成员变化前后的数据，需要注意的是成员变化的通知。

例如，通过

```
var rootCollectionsFetchResult = PHCollection.fetchTopLevelUserCollectionsWithOptions
```

获取所有用户建立的相册和文件夹，在 `photoLibraryDidChange(changeInfo: PHChange!)` 中通过以下方法获得 `PHFetchResultChangeDetails` 对象。

```
let fetchChangeDetails = changeInstance.changeDetailsForFetchResult(rootCollectionsF
```

`fetchChangeDetails.changedObject` 返回一组其内容或元数据发生变化的成员，返回的成员是更新后的成员对象。当用户对某个文件夹内的相册或子文件夹进行添加、删除和编辑操作即文件夹的内容而不是文件夹本身的属性发生变化时，通知中会该变化的信息吗？实际上只有在文件夹中添加相册或子文件夹时才会对 `fetchChangeDetails.changedObject` 中有所反应，而删除成员或是修改元数据等操作都不会在通知有所反应，你需要使用其他手段来跟踪变化。

➕ 推荐拓展阅读 (/sign\_in)

© 著作权归作者所有

△O×□

¥ 打赏支持



♡ 喜欢 | 50

👤 分享到微博    🐧 分享到微信  
更多分享 ▼

20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)



米苏芊 (/users/49e5ef13b7d3)

2楼 · 2016.03.09 23:31 (/p/42e5d2f75452/comments/1670346#comment-1670346)

非常棒的总结介绍

♡ 喜欢(0)

回复



Will\_Coder (/users/989241c1852f)

3楼 · 2016.03.24 16:14 (/p/42e5d2f75452/comments/1817984#comment-1817984)



(/users/989241c1852f)

♡ 喜欢(0)

回复



四心儿 (/users/ba6da4472da4)

5楼 2016.04.19 18:26 (/p/42e5d2f75452/comments/2098852#comment-2098852)

请问怎么用PHPhotoLibraryChange删除本地照片或视频，下面是我的代码，但是删除不了

```
PHPhotoLibrary.sharedPhotoLibrary().performChanges( {  
let AssetToDelete = PHAsset.fetchAssetsWithALAssetURLs(self.pathV, options: nil)  
PHAssetChangeRequest.deleteAssets(AssetToDelete)}, completionHandler: nil )
```

♡ 喜欢(0)

回复

seedante (/users/7ba5d9065301): @四心儿 (/users/ba6da4472da4) 你没有收到删除通知啥的?

2016.04.19 18:52 (/p/42e5d2f75452/comments/2099096#comment-2099096)

回复

✎ 添加新回复



米苏芊 (/users/49e5ef13b7d3)

5楼 2016.05.31 15:16 (/p/42e5d2f75452/comments/2592104#comment-2592104)

albumChangeRequest!.addAssets(NSFastEnumeration) 这个现在是 NSFastEnumeration 类型，这个怎么弄，不太了解

♡ 喜欢(0)

回复



我是卖报滴小行家 (/users/b7c8a2e1f998)

6楼 2016.07.22 14:50 (/p/42e5d2f75452/comments/3232630#comment-3232630)

Photos框架能不能获取文件的有多少MB，像ALAsset的[[model.asset defaultRepresentation] size]这样的，，

♡ 喜欢(0)

回复

seedante (/users/7ba5d9065301): @我是卖报滴小行家 (/users/b7c8a2e1f998) 看了下 API，没有公开这个接口。这里有个比较曲折的方法，通过获取的 imageData 来计算：<http://stackoverflow.com/questions/26549938/how-can-i-determine-file-size-on-disk-of-a-video-phasset-in-ios8/38059090#38059090> (<http://stackoverflow.com/questions/26549938/how-can-i-determine-file-size-on-disk-of-a-video-phasset-in-ios8/38059090#38059090>)。

2016.07.22 15:12 (/p/42e5d2f75452/comments/3232962#comment-3232962)

回复

我是卖报滴小行家 (/users/b7c8a2e1f998): @seedante (/users/7ba5d9065301) 3Q 😊

2016.07.23 11:37 (/p/42e5d2f75452/comments/3245618#comment-3245618)

回复

✎ 添加新回复



搬砖公 (/users/2d3bab7e50f9)

7楼 · 2016.09.02 20:46 (/p/42e5d2f75452/comments/3982712#comment-3982712)

请问如何获取原始图片,我得到的都是缩略图,质量比较差

♡ 喜欢(0)

回复

seedante (/users/7ba5d9065301): @搬砖公 (/users/2d3bab7e50f9) 看完「请求图像」这一节。

2016.09.02 21:21 (/p/42e5d2f75452/comments/3983311#comment-3983311)

回复

✎ 添加新回复



旧夏2014 (/users/54cc1a5fc8b8)

8楼 · 2016.09.24 16:57 (/p/42e5d2f75452/comments/4388790#comment-4388790)

想要读出来相册里图片的顺序跟系统相册里图片的顺序一样怎么设置option呢, 现在知道的有@“modificationDate”和@“creationDate”

♡ 喜欢(0)

回复



Mustard\_iOS (/users/dc1f6f07e145)

9楼 · 2016.10.18 08:56 (/p/42e5d2f75452/comments/4912297#comment-4912297)

resizeMode为什么不能为none。只是因为你没有算上屏幕的scale。

♡ 喜欢(0)

回复

seedante (/users/7ba5d9065301): @Mustard\_iOS (/users/dc1f6f07e145) resizeMode 为 .None 时 UIImageManager 不会对源图像进行缩放

2016.10.18 12:38 (/p/42e5d2f75452/comments/4917717#comment-4917717)

回复

Mustard\_iOS (/users/dc1f6f07e145): @seedante (/users/7ba5d9065301) 那当option为nil的时候, 这个resizeMode是啥? 😞😞

2016.10.18 15:58 (/p/42e5d2f75452/comments/4922539#comment-4922539)

回复



简书  
手机  
版下载

seedante (/users/7ba5d9065301): @Mustard\_iOS (/users/dc1f6f07e145) 不指定 option 效果等同 resizeMode 为 .None。

2016.10.18 17:59 (/p/42e5d2f75452/comments/4925442#comment-4925442)

回复

还有 2 条评论, 展开查看

添加新回复



manajay (/users/089cf0c28ed5)

你好 我做分片上次视频的功能, 使用 photokit 想获取视频的路径, 但是发现URL 没用 不能用 fileHandler 获取文件数据, 那么, 有什么方法 可以做到吗, 比如localIdentifier是会动态改变的吗

喜欢(0)

回复

seedante (/users/7ba5d9065301): @Manajay (/users/089cf0c28ed5) 没弄过视频, 不清楚。至于 localIdentifier, 我看我的文章发现也没有明确指出这个问题, 不过你试验下应该就能知道是否是动态变化的了。

2016.10.24 12:34 (/p/42e5d2f75452/comments/5065356#comment-5065356)

回复

manajay (/users/089cf0c28ed5): @seedante (/users/7ba5d9065301) 嗯好的 谢谢

2016.10.24 18:00 (/p/42e5d2f75452/comments/5072829#comment-5072829)

回复

添加新回复

登录后发表评论 (/sign\_in)

被以下专题收入, 发现更多相似内容:



**iOS学习 (/collection/1332c736fe39)**

学习从点滴开始! (PS: 拒绝部分投稿的文章仅仅是由于专题内已收录相关知识点的文章, 并非是投稿的文章技术含量不够好, 望谅解.)

4749篇文章 (/collection/1332c736fe39) · 5609人关注

添加关注 (/sign\_in)



**快速入门iOS (/collection/11adc632c888)**

iOS, Swift, OC, 架构

添加关注 (/sign\_in)

(/collection/11adc632c888) · 3943人关注



**iOS-swift (/collection/31622a1d4959)**

iOS

(/collection/31622a1d4959) · 2563人关注

[+ 添加关注 \(/sign\\_in\)](#)