

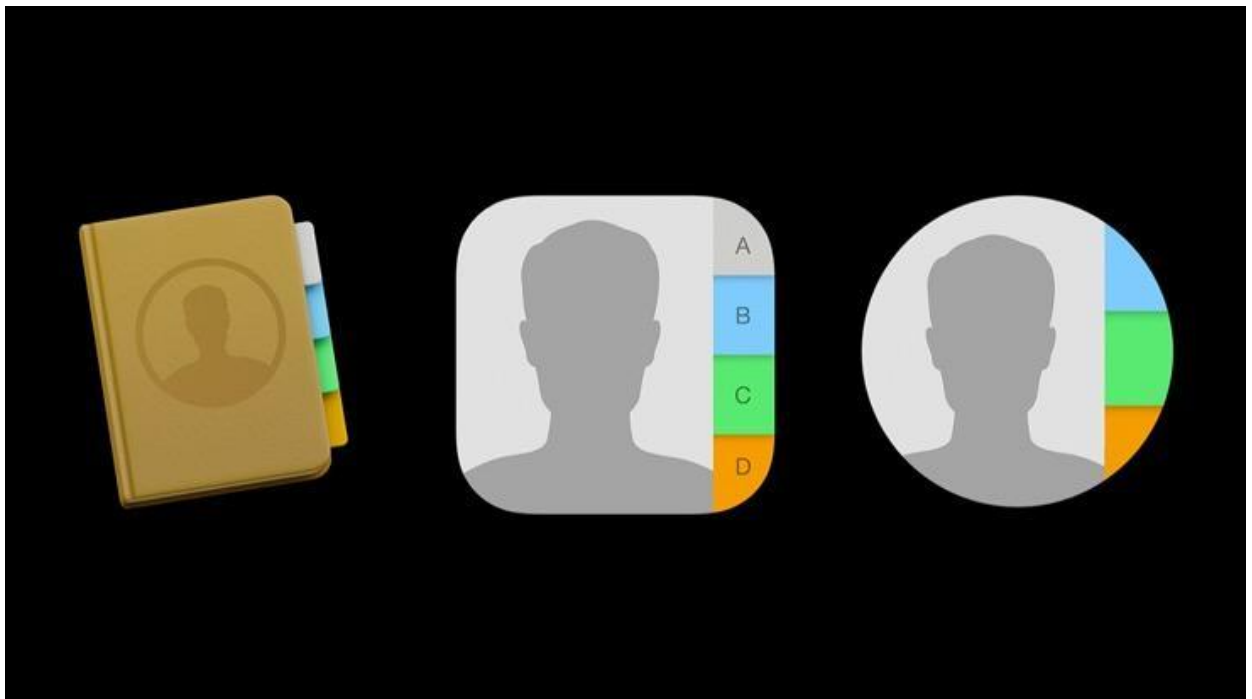


作者 Two_Seven (/users/8e9c5d7530a6) 2016.05.16 17:08
写了9054字，被147人关注，获得了199个喜欢
(/users

+ 添加关注 (/sign_in)

iOS Contacts framework 通讯录新框架，取代AddressBook

字数1409 阅读283 评论1 喜欢6



Contacts框架

刚刚查了一下，iOS 9系统的市场份额大概占到了80%左右，所以如果你能够说服老板放弃剩下的20%左右的用户的话（我感觉老板不会被你忽悠），你现在就可以只使用Contacts这个新框架了，否则的话就还是乖乖的用AddressBook吧，或者通过判断系统版本号来适配不同的手机。

好吧，现在开始进入正题，这个新框架中将要使用最多的应该就是CNContactStore这个类了，官方对其的定义是一个线程安全的，用来执行查询和修改操作的类。此外官方还针对于这个类的使用给出了几点小建议。

1.只去查询那些你需要的联系人属性。

2.当查询所有联系人并缓存结果时，首先查询所有联系人的identifier,然后通过

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

(/collections)



(/apps)

3.想取出多个联系人信息时，可以先收集联系人的唯一identifier，然后通过identifier来查询所需要的详细的联系人信息。

4.如果缓存了联系人信息，那么当接收到CNCContactStoreDidChangeNotification时你需要刷新数据。

5.CNCContactStore的查询方法中执行了I/O操作，所以应该避免在主线程中执行。

在项目中，首先我们可以调用授权状态的查询方法来确定应用的授权状态。

```
CNAuthorizationStatus status = [CNCContactStore authorizationStatusForEntityType:CNEntityTypeContacts];
```

NotDetermined：这个状态说明用户暂未决定是否允许访问联系人数据库。当应用第一次安装在设备上时将处于此状态。

Restricted：这个状态说明应用不仅不能够访问联系人数据，并且用户也不能在设置中改变这个状态。这个状态是某些被激活的限制所导致的（比如说家长控制）。

Denied：这个状态说明用户不允许应用访问联系人数据。这个状态只能够被用户改变。

Authorized：这个状态是所有应用都希望拥有的，这表明应用能够自由访问联系人数据库，然后根据联系人数据来处理某些任务。

授权状态一共是上面的四种，我们需要分别对不同的授权状态来做出相应的操作。首先如果是**Authorized**的话，那么一切好说，接下来直接进行相应的通讯录操作就好了，而如果是**NotDetermined**的话，就调用一下请求授权的方法（下面会详细说明），至于剩下的两个方法，则只能弹窗提示用户去设置里面修改权限了。

那么接下来，我们假设调用上面的方法，得到的结果是**NotDetermined**，我们就需要调用下面的方法来请求授权，如果用户不同意，那就弹窗问他，不给我通讯录授权我怎么服务你啊（在这点上脉脉做的比较狠😏）。

```
CNCContactStore *store = [[CNCContactStore alloc] init];
[store requestAccessForEntityType:CNEntityTypeContacts completionHandler:^(BOOL granted, NSError *) {
    // ...
}];
```

如果顺利的话，到这里我们就已经拿到用户的授权了，那么接下来就可以读取用户数据了。首先，我们需要创建一个

CNCContactFetchRequest对象。这里简单说一下

predicate 和 **keysToFetch** 这两个参数，其中的 **predicate** 谓词参数，必须是由 **CNCContact+Predicates** 生成的。至于 **keysToFetch** 参数，个人理解是由两部分组成的，其中一部分可以去 **CNCContact.h** 的最下面查，另一部分可以查询 **CNCContactFormatter.h**。下面是我写的一小段示例代码：

A



简
(/)

☰

(/collections)

☰

(/apps)

```
CNContactFetchRequest *request = [[CNContactFetchRequest alloc] initWithKeysToFetch:
    NSArray *keys = @[CNContactFirstNameKey, CNContactLastNameKey, CNContactPhoneNumberKey];
    NSError *error = nil;
    self.array = [NSMutableArray array];
    [store enumerateContactsWithFetchRequest:request error:&error usingBlock:^(CNContact *contact, NSUInteger index, BOOL *stop, NSError *error) {
        [self.array addObject:contact];
    }];
```

如果没什么意外情况的话，所需要的通讯录信息都已经存在这个数组里面了，需要注意的是这个方法的结果不是一次性返回的，这个可以通过监控stop参数来确定。（此外就是在这里我有一个疑问，希望理解的同学帮我解释一下。我在官方文档中看到一种说法就是其实只要调用查询方法，系统就会自动去请求授权。所以我对上面提到的那个请求授权的方法的调用时机不是完全的理解）。

那么最后，当我们获取到通讯录信息之后，可能会将数据暂存下来，这样就涉及到了一个数据刷新的问题，这个问题可以通过添加

CNContactStoreDidChangeNotification监听来解决，需要注意的是这个监听的响应次数是和通讯录中修改信息的条数相匹配的，所以一定要做好相应的处理。

到这里简单的获取通讯录信息功能应该实现了，最后说一下iOS8之后是可以直接从应用内跳转到设置页面来修改通讯录权限的，所以还是做一下系统适配的好，不然也许用户嫌麻烦就不去自己改了。

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

最后是作者唠叨时间，感谢大家对我的支持，尤其是上一篇文章大家对我的肯定，还是老样子，如果大家感觉对你有一定的帮助，就关注加喜欢吧，就当是小小满足一下我的虚荣心，我们一起进步。

➕ 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

A

第3页 共4页
(/sign_in)

♡ 喜欢 | 6

🔗 分享到微博 🗨 分享到微信 更多分享 ⏴ 4:48



简书 (/)

1条评论 (按时间正序 · 按时间倒序 · 按喜欢排序)

登录 (/sign_in) 注册 (/sign_up)



(/collections)



品尝一夏 (/users/12130f301648)

2楼 · 2016-05-18 09:18 (/p/8de5a30cdda4/comments/2430349#comment-2430349)



(/apps)

作者你好！请问在你使用系统通讯录过程中，有没有遇到过这样的问题：iOS9.x下，查看联系人详情，点击“共享联系人”之后，会顶掉当前页的导航栏，导致页面无法返回只能强制关掉。是iOS9.x不兼容之前的AddressBook框架的原因还是其他的？希望作者可以看到指点一丢，先行谢过。

喜欢(0)

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：



首页投稿 (/collection/bDHhpK)

玩转简书的第一步，从这个专题开始。想上首页热门榜么？好内容想被更多人看到 (/collection/bDHhpK) 来投稿吧！如果被拒也不要灰心哦~入选文章会进一个队...

56369篇文章 (/collection/bDHhpK) · 100003人关注



添加关注 (/sign_in)



程序员 (/collection/NEt52a)

简书程序员大本营 投稿须知： 1.本专题仅收录与程序有关的文章。 2.请在代码框里 (/collection/NEt52a) 写代码，尽量保证可看性。关注简书官...

13179篇文章 (/collection/NEt52a) · 89798人关注



添加关注 (/sign_in)



iOS Developer (/collection/3233d1a249ca)

分享 iOS 开发的知识，解决大家遇到的问题，讨论iOS开发的前沿，欢迎大家投稿 (/collection/3233d1a249ca) ~

6018篇文章 (/collection/3233d1a249ca) · 18795人关注



添加关注 (/sign_in)

A

