

首页 > iOS开发

iOS 硬件授权检测：定位服务、通讯录、日历、提醒事项、照片、蓝牙共享、麦克风、相机等

2015-12-14 10:06 编辑: lansekuangtu 分类: iOS开发 来源: 金田 的博客

1 1377

[通讯录](#) [照片](#) [日历](#) [iOS](#) [硬件授权检测](#) [定位服务](#) [提醒事项](#)

招聘信息: iOS开发工程师



作者: 金田

iOS系统版本的不断升级的前提，伴随着用户使用设备的安全性提升，iOS系统对于App需要使用的硬件限制也越来越严格，App处理稍有不妥，轻则造成功能不可用用户还不知道，重则会造成App Crash。

当用户在App启动时，看到弹出来的一条条“XXX 请求访问您的位置”“XXX 请求访问您的通讯录”“XXX 请求访问您的日历”“XXX 请求访问您的摄像头”等一系列消息时，用户觉得不耐烦的同时，也会由于一时的安全考虑而把相应的功能给屏蔽掉，这还只是开始，当用户真正在使用对应功能的时候，就会出现一连续的奇怪现象，比如数据显示异常：明明通讯录里面有信息，却总是加载不出数据；有的甚至是直接Crash。

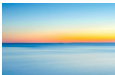

下面，笔者将会综合性地把上述硬件的授权检测，一一地详细列出，并给出相关示例代码：

1、定位服务

相关框架

#import

热门资讯

 如何让iOS 保持界面流畅？这些技巧你知道吗
点击量 13485 【资源集合】94个iOS 开发资源推荐，帮你加
点击量 8525 苹果Swift编程语言开 源 有望进一步普及
点击量 6401 对 Xcode 菜单选项的 详细探索（干货）
点击量 6366 Swift 3 API 设计准则
点击量 5745 iOS 事件处理机制与图 像渲染过程
点击量 5617 一款Loading动画的实 现思路（一）：拆分复
点击量 5534 Swift中编写单例的正 确方式
点击量 5253 经验之谈！35岁程序员 的独家面试经历
点击量 5116 必读！结合实际问题的 Bitcode 适配指南
点击量 4922

综合评论

mark
121 评论了 【资源集合】94个iOS开发 资源推荐，帮你加速应用...为什么这么好的帖子却没有评论， 虽然我是新手，好多看不懂
goldWave 评论了 扒一扒知乎上的帖子 ——“为什么有些大公司技术弱爆了...mark
自来也大人 评论了 Objective-C 引用 计数：不讲用法，只说原...

检测方法

```
+ (CLLocationAuthorizationStatus)authorizationStatus
```

相关返回参数

```
//用户尚未做出选择

kCLLocationAuthorizationStatusNotDetermined = 0,


// 未授权，且用户无法更新，如家长控制情况下

kCLLocationAuthorizationStatusRestricted,


// 用户拒绝App使用

kCLLocationAuthorizationStatusDenied,


// 总是使用

kCLLocationAuthorizationStatusAuthorizedAlways NS_ENUM_AVAILABLE(NA, 8_0),


// 按需使用

kCLLocationAuthorizationStatusAuthorizedWhenInUse NS_ENUM_AVAILABLE(NA, 8_0),


// 已授权

kCLLocationAuthorizationStatusAuthorized
```

参考代码

```
__block void (^checkLocationAuth)(void) = ^{

    CLLocationAuthorizationStatus authStatus = [CLLocationManager authorizationStatus];


    if (CLLocationAuthorizationStatusAuthorized == authStatus) {

        //授权成功，执行后续操作

    }else {

        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{

            checkLocationAuth();

        });

    }

};

checkLocationAuth();
```

2、通讯录

```
#import
```

检测方法

```
ABAuthorizationStatus ABAddressBookGetAuthorizationStatus(void)
```

授权状态

```
kABAuthorizationStatusNotDetermined = 0, // 未进行授权选择
```

说白了还是自学能力的高低

Grahancoxom 评论了 培训机构毕业的程序员被歧视的背后逻辑...

信息就是这都是情怀，钱没多少的。
soniceme 评论了 实际经历告诉你，写一本技术书能赚多少钱...

nice
shiyao357 评论了 UIScrollView 实践经验

mark
Grahancoxom 评论了 【资源集合】94个iOS开发资源推荐，帮你加速应用...

呵呵，国内前几的支付公司几年前连请求都没加密的，也见过好多公司用户名
soniceme 评论了 扒一扒知乎上的帖子——“为什么有些大公司技术弱爆了...

不明觉厉
zmings 评论了 尽量不要在viewWillAppear:方法...

很有帮助呀，看来面试之前确实要做很多准备工作，才能更有把握
pxl_accp 评论了 经验之谈！35岁程序员的独家面试经历...

相关帖子

- 崩溃解决
- 求助xcode7.2
- 纠结：我不敢带男友出去参加聚会
- 江湖救急！字体适配屏幕问题
- 江湖救急！字体适配屏幕问题
- 江湖救急！字体适配屏幕问题
- cocos-2d-js-live 版手机pad运行错误
- 自定义UIImageView的旋转问题
- cocosstudio资源创建的意见

```
kABAuthorizationStatusRestricted,          // 未授权，且用户无法更新，如家长控制情况下
kABAuthorizationStatusDenied,              // 用户拒绝App使用
kABAuthorizationStatusAuthorized           // 已授权，可使用
```

参考代码

```
__block void (^checkAddressBookAuth)(void) = ^{
    ABAuthorizationStatus authStatus = ABAddressBookGetAuthorizationStatus();

    if (kABAuthorizationStatusAuthorized == authStatus) {
        //授权成功，执行后续操作
    }else {
        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{
            checkAddressBookAuth();
        });
    }
};

checkAddressBookAuth();
```

3、日历/提醒事项授权

#import

检测方法

+ (EKAuthorizationStatus)authorizationStatusForEntityType:(EKEntityType)entityType

参数类型

EKEntityTypeEvent, //事件

EKEntityTypeReminder//提醒

授权状态

EKAuthorizationStatusNotDetermined = 0, // 未进行授权选择

EKAuthorizationStatusRestricted, // 未授权，且用户无法更新，如家长控制情况下

EKAuthorizationStatusDenied, // 用户拒绝App使用

EKAuthorizationStatusAuthorized, // 已授权，可使用

参考代码

```
__block void (^checkCalanderAuth)(void) = ^{
    EKAuthorizationStatus authStatus = [EKAuthorizationStatus authorizationStatusForEntityType:EKEntityTypeEvent];

    if (EKAuthorizationStatusAuthorized == authStatus) {
        //授权成功，执行后续操作
    }else {
```

微博



CocoaChina

加关注

【接口编程那些事（或者面向协议编程）】接口是一系列可调方法的集合。何为接口编程？接口编程是指当写一个函数或一个方法时，我们应该更加关注具体的接口，而不是实现类。在OC中，接口又可以理解为Protocol，面向接口编程又可以理解为面向Protocol编程，或者面向协议编程。<http://t.cn/R4wN5Fq>




12月14日 21:58 转发(15) | 评论(5)

【成为更优秀关卡设计师需要养成的4个习惯】有时候我会遇到一些总是希望将自己的工作做得更好的设计师



```
        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5 * NSEC_PER_SEC)), dispatch_get_main_queue(),
            ^{
                checkCalanderAuth();
            });
    }
};

checkCalanderAuth();
```



4、照片库授权

#import

检测方法

+ (ALAuthorizationStatus)authorizationStatus;

授权状态

ALAuthorizationStatusNotDetermined = 0, // 未进行授权选择

ALAuthorizationStatusRestricted, // 未授权，且用户无法更新，如家长控制情况下

ALAuthorizationStatusDenied, // 用户拒绝App使用


ALAuthorizationStatusAuthorized, // 已授权，可使用

参考代码

```
__block void (^checkAssetLibraryAuth)(void) = ^{
    ALAuthorizationStatus authStatus = [ALAuthorizationStatus authorizationStatus];

    if (ALAuthorizationStatusAuthorized == authStatus) {
        //授权成功，执行后续操作
    }else {
        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5 * NSEC_PER_SEC)), dispatch_get_main_queue(),
            ^{
                checkAssetLibraryAuth();
            });
    }
};

checkAssetLibraryAuth();
```



5、蓝牙授权状态检测

#import

检测方法

+ (CBPeripheralManagerAuthorizationStatus)authorizationStatus;

授权状态

CBPeripheralManagerAuthorizationStatusNotDetermined = 0, // 未进行授权选择

```

CBPeripheralManagerAuthorizationStatusRestricted,          // 未授权，且用户无法更新，如家长控制情况下

CBPeripheralManagerAuthorizationStatusDenied,             // 用户拒绝App使用

CBPeripheralManagerAuthorizationStatusAuthorized,         // 已授权，可使用

```

参考代码

```

__block void (^checkPeripheralAuth)(void) = ^{
    CBPeripheralManagerAuthorizationStatus authStatus = [CBPeripheralManagerAuthorizationStatus aut

    if (CBPeripheralManagerAuthorizationStatusAuthorized == authStatus) {
        //授权成功，执行后续操作
    }else {
        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5 * NSEC_PER_SEC)), dispatch_ge
            checkPeripheralAuth());
    }
};

checkPeripheralAuth();

```

6、摄像头授权状态检测

```
#import
```

检测方法

```
+ (AVAuthorizationStatus)authorizationStatusForMediaType:(NSString *)mediaType;
```

授权状态

```

AVAuthorizationStatusNotDetermined = 0, // 未进行授权选择

AVAuthorizationStatusRestricted,        // 未授权，且用户无法更新，如家长控制情况下

AVAuthorizationStatusDenied,            // 用户拒绝App使用

AVAuthorizationStatusAuthorized,        // 已授权，可使用

```

参考代码

```

__block void (^checkCameraAuth)(void) = ^{
    AVAuthorizationStatus authStatus = [AVAuthorizationStatus authorizationStatusForMediaType:AVMed

    if (AVAuthorizationStatusAuthorized == authStatus) {
        //授权成功，执行后续操作
    }else {
        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5 * NSEC_PER_SEC)), dispatch_ge
            checkCameraAuth());
    }
};

```

```
    });  
}  
};  
  
checkCameraAuth();
```

7、麦克风授权状态检测

#import

检测方法

```
- (AVAudioSessionRecordPermission)recordPermission;
```

授权状态

```
AVAudioSessionRecordPermissionUndetermined,    // 用户尚未被请求许可。
```

```
AVAudioSessionRecordPermissionDenied,          // 用户已被要求并已拒绝许可。
```

```
AVAudioSessionRecordPermissionGranted,         // 用户已被要求并已授予权限。
```

参考代码

```
__block void (^checkRecordPermission)(void) = ^{  
    AVAudioSessionRecordPermission authStatus = [[AVAudioSession sharedInstance] recordPermission];  
  
    if (AVAudioSessionRecordPermissionGranted == authStatus) {  
        //授权成功, 执行后续操作  
    }else {  
        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{  
            checkRecordPermission();  
        });  
    }  
};  
  
checkRecordPermission();
```



微信扫一扫

订阅每日移动开发及APP推广热点资讯
公众号: CocoaChina

我要投稿

收藏文章

分享到:

17

上一篇：源码推荐(11.30B)：最流行的 UINavigationController，新手引导页面效果

下一篇：知其然亦知其所以然--NSOperation并发编程

相关资讯

源码推荐(12.14)：iOS mansory纯代码自动计算cell高度，手把手教你如何分析 iOS 系统栈 crash

36氪客户端4.0版本开发架构总结：第三方分享认证模块的

【译】自动更新订阅IAP浅谈（设置和测试）

iOS 事件处理机制与图像渲染过程

【资源集合】94个iOS开发资源推荐，帮你加速应用开发

利用OC的消息转发机制实现多重代理

iOS新建项目架构规范

iOS微信安装包瘦身

如何跳到系统设置里的各种设置界面



我来说两句



你怎么看？快来评论一下吧！

发表评论

所有评论（1）



jas_

2015-12-05 02:03:29

刚好用到~ mark

0 0 回复

[关于我们](#) [商务合作](#) [联系我们](#) [合作伙伴](#)

北京触控科技有限公司版权所有

©2015 Chukong Technologies, Inc.

京ICP备 11006519号 京ICP证 100954号 京公网安备11010502020289



京网文[2012]0426-138号