

# iOS研发助手DoraemonKit技术实现（一）

(/apps/  
utm\_so  
banner



景铭巴巴 (/u/c3c893a27097) [+关注](#)

💎 6.4 2018.10.28 21:45\* 字数 3303 阅读 7585 评论 18 喜欢 52

(/u/c3c893a27097)

## 一、前言

一个比较成熟的App，经历了多个版本的迭代之后，为了方便调式和测试，往往会积累一些工具来应付这些场景。最近我们组就开源了一款适用于iOS App线下开发、测试、验收阶段，内置在App中的工具集合。使用DoraemonKit，你无需连接电脑，就可以对于App的信息进行快速的查看和修改。一键接入、使用方便，提高开发、测试、视觉同学的工作效率，提高我们App上线的完整度和稳定性。

目前DoraemonKit拥有的功能大概分为以下几点：

1. 常用工具：App信息展示，沙盒浏览、MockGPS、H5任意门、子线程UI检查、日志显示。
2. 性能工具：帧率监控、CPU监控、内存监控、流量监控、自定义监控。
3. 视觉工具：颜色吸管、组件检查、对齐标尺。
4. 业务专区：支持业务测试组件接入到DoraemonKit面板中。

拿我们App接入效果如下：

(https:/  
yex.you  
slot=3(  
613e-4  
fe328e  
ttd\_eA  
FEY\_W\  
fdwdH  
click.yc  
613e-4  
fe328e  
391515





上面两行是业务线自定义的工具，接入方可以自定义。除此之外都是内置工具集合。

因为里面功能比较多，大概会分三篇文章介绍DoraemonKit的使用和技术实现，这是第一篇主要介绍常用工具集中的几款工具实现。

## 二、技术实现

### 2.1：App信息展示



我们要看一些手机信息或者App的一些基本信息的时候，需要到系统设置去找，比较麻烦。特别是权限信息，在我们app装的比较多时候，我们很难快速找到我们app的权限信息。而这些信息从代码角度都是比较容易获取的。我们把感兴趣的信息列表出来直接查看，避免了去手机设置里查看或者查看源代码的麻烦。

## 获取手机型号

我们从手机设置里面是找不到我们的手机具体是哪一款的文字表述的，比如我的手机是iphone8 Pro，在手机型号里面显示的是MQ8E2CH/A。对于iPhone不熟悉的人很难从外表对iphone进行区分。而手机型号，我们从代码角度就很好获取。

```
+ (NSString *)iphoneType{
    struct utsname systemInfo;
    uname(&systemInfo);
    NSString *platform = [NSString stringWithCString:systemInfo.machine encoding:NSUTF8StringEncoding];

    //iPhone
    if ([platform isEqualToString:@"iPhone1,1"]) return @"iPhone 1G";
    ...
    //其他对应关系请看下面对应表
    return platform;
}
```

iPhone设备类型与通用手机类型一一对应关系表

设备类型	通用类型
iPhone1,1	iPhone 1G
iPhone1,2	iPhone 3G
iPhone2,1	iPhone 3GS
iPhone3,1	iPhone 4
iPhone3,2	iPhone 4
iPhone4,1	iPhone 4S
iPhone5,1	iPhone 5
iPhone5,2	iPhone 5
iPhone5,3	iPhone 5C
iPhone5,4	iPhone 5C
iPhone6,1	iPhone 5S
iPhone6,2	iPhone 5S

iPhone7,1	iPhone 6 Plus
iPhone7,2	iPhone 6
iPhone8,1	iPhone 6S
iPhone8,2	iPhone 6S Plus
iPhone8,4	iPhone SE
iPhone9,1	iPhone 7
iPhone9,3	iPhone 7
iPhone9,2	iPhone 7 Plus
iPhone9,4	iPhone 7 Plus
iPhone10,1	iPhone 8
iPhone10,4	iPhone 8
iPhone10,2	iPhone 8 Plus
iPhone10,5	iPhone 8 Plus
iPhone10,3	iPhone X
iPhone10,6	iPhone X
iPhone11,8	iPhone XR
iPhone11,2	iPhone XS
iPhone11,4	iPhone XS Max
Phone11,6	iPhone XS Max

## 获取手机系统版本

```
//获取手机系统版本
NSString *phoneVersion = [[UIDevice currentDevice] systemVersion];
```

## 获取App BundleId

一个app分为测试版本、企业版本、appStore发售版本，每一个app长得都一样，如何对他们进行区分呢，那就要用到BundleId这个属性了。

```
//获取bundle id
NSString *bundleId = [[NSBundle mainBundle] bundleIdentifier];
```

## 获取App 版本号

```
//获取App版本号
NSString *bundleVersionCode = [[[NSBundle mainBundle]infoDictionary] object
ForKey:@"CFBundleVersion"];
```

## 权限信息查看

当我们发现App运行不正常，比如无法定位，网络一直失败，无法收到推送信息等问题的时候，我们第一个反应就是去手机设置里面去看我们app相关的权限有没有打开。DoraemonKit集成了对于地理位置权限、网络权限、推送权限、相机权限、麦克风权限、相册权限、通讯录权限、日历权限、提醒事项权限的查询。

由于代码比较多，这里就不一一贴出来了。大家可以去DoraemonKit/Core/Plugin/AppInfo中自己去查看。这里讲一下，权限查询结果几个值的意义。

- NotDetermined => 用户还没有选择。
- Restricted => 该权限受限，比如家长控制。
- Denied => 用户拒绝使用该权限。
- Authorized => 用户同意使用该权限。

## 2.2：沙盒浏览

### 沙盒浏览器.png

以前如果我们要去查看App缓存、日志信息，都需要访问沙盒。由于iOS的封闭性，我们无法直接查看沙盒中的文件内容。如果我们要去访问沙盒，基本上有两种方式，第一种使用Xcode自带的工具，从Windows-->Devices进入设备管理界面，通过Download Container的方式导出整个app的沙盒。第二种方式，就是自己写代码，访问沙盒中指定文件，然后使用NSLog的方式打印出来。这两种方式都比较麻烦。

DoraemonKit给出的解决方案：就是自己做一个简单的文件浏览器，通过NSFileManager对象对沙盒文件进行遍历，同时支持对于文件和文件夹的删除操作。对于文件支持本地预览或者通过airdrop的方式或者其他分享方式发送到PC端进行更加细致的操作。

怎么用NSFileManager对象遍历文件和删除文件这里就不说了，大家可以参考DoraemonKit/Core/Plugin/Sandbox中的代码。这里讲一下：如何将手机中的文件快速上传到Mac端？刚开始我们还绕了一点路，我们在手机端搭了一个微服务，mac通过浏览器去访问它。后来和同事聊天的时候知道了UIActivityViewController这个类，可以十分便捷地吊起系统分享组件或者是其他注册到系统分享组件中的分享方式，比如微信、钉钉。实现代码非常简单，如下所示：

```
- (void)shareFileWithPath:(NSString *)filePath{

    NSURL *url = [NSURL URLWithString:filePath];
    NSArray *objectsToShare = @[url];

    UIActivityViewController *controller = [[UIActivityViewController alloc] initWithActivityItems:objectsToShare applicationActivities:nil];
    NSArray *excludedActivities = @[UIActivityTypePostToTwitter, UIActivityTypePostToFacebook,
                                     UIActivityTypePostToWeibo,
                                     UIActivityTypeMessage, UIActivityTypeMail,
                                     UIActivityTypePrint, UIActivityTypeCopyToPasteboard,
                                     UIActivityTypeAssignToContact, UIActivityTypeSaveToCameraRoll,
                                     UIActivityTypeAddToReadingList, UIActivityTypePostToFlickr,
                                     UIActivityTypePostToVimeo, UIActivityTypePostToTencentWeibo];
    controller.excludedActivityTypes = excludedActivities;

    [self presentViewController:controller animated:YES completion:nil];
}
```

## 2.3: MockGPS

mockGPS.png

我们有些业务会根据地理位置不同，而有不同的业务处理逻辑。而我们开发或者测试，当然不可能去每一个地址都测试一遍。这种情况下，测试同学一般会找到我们让我们手动改掉系统获取经纬度的回调，或者修改GPX文件，然后再重新打一个包。这样也非常麻烦。

DoraemonKit给出的解决方案：提供一套地图界面，支持在地图中滑动选择或者手动输入经纬度，然后自动替换掉我们App中返回的当前经纬度信息。这里的难点是如何不需要重新打包自动替换掉系统返回的当前经纬度信息？

CLLocationManager的delegate中有一个方法如下：

```
/*
 * locationManager:didUpdateLocations:
 *
 * Discussion:
 *   Invoked when new locations are available. Required for delivery of
 *   deferred locations. If implemented, updates will
 *   not be delivered to locationManager:didUpdateToLocation:fromLocation:
 *
 *   locations is an array of CLLocation objects in chronological order.
 */
- (void)locationManager:(CLLocationManager *)manager
    didUpdateLocations:(NSArray<CLLocation *> *)locations API_AVAILABLE(ios(6.0), macos(10.9));
```

我们通常是在这个函数中获取当前系统的经纬度信息。我们如果想要没有侵入式的修改这个函数的默认实现方式，想到的第一个方法就是Method Swizzling。但是真正在实现过程中，你会发现Method Swizzling需要当前实例和方法，方法是-

```
(void)locationManager:(CLLocationManager *)manager
didUpdateLocations:(NSArray<CLLocation *> *)locations
```

我们有了，但是实例，每一个app都有自己的实现，无法做到统一处理。我们就换了一个思路，如何能获取该实现了该定位方法的实例呢？就是使用Method Swizzling Hook住CLLocationManager的setDelegate方法，就能获取具体是哪一个实例实现了- (void)locationManager:(CLLocationManager \*)manager didUpdateLocations:(NSArray<CLLocation \*> \*)locations 方法。

具体方法如下：

第一步：生成一个CLLocationManager的分类CLLocationManager(Doraemon),在这个分类中，实现- (void)doraemon\_swizzleLocationDelegate:(id)delegate这个方法，用来进行方法交换。

```
- (void)doraemon_swizzleLocationDelegate:(id)delegate {
    if (delegate) {
        //1、让所有的CLLocationManager的代理都设置为[DoraemonGPSPMocker sharedInstance]，让他做中间转发
        [self doraemon_swizzleLocationDelegate:[DoraemonGPSPMocker sharedInstance]];
        //2、绑定所有CLLocationManager实例与delegate的关系，用于[DoraemonGPSPMocker sharedInstance]做目标转发用。
        [[DoraemonGPSPMocker sharedInstance] addLocationBinder:self delegate:delegate];

        //3、处理[DoraemonGPSPMocker sharedInstance]没有实现的selector，并且给用户提示。
        Protocol *proto = objc_getProtocol("CLLocationManagerDelegate");
        unsigned int count;
        struct objc_method_description *methods = protocol_copyMethodDescriptionList(proto, NO, YES, &count);
        NSMutableArray *array = [NSMutableArray arrayWithCapacity:count];
        for(unsigned i = 0; i < count; i++)
        {
            SEL sel = methods[i].name;
            if ([delegate respondsToSelector:sel]) {
                if (![DoraemonGPSPMocker sharedInstance] respondsToSelector:sel) {
                    NSLog(@"你在Delegate %@ 中所使用的SEL %@, 暂不支持，请联系DoraemonKit开发者", delegate, sel);
                }
            }
        }
        free(methods);
    }
    else{
        [self doraemon_swizzleLocationDelegate:delegate];
    }
}
```



在这个函数中主要做了三件事情，1、将所有的定位回调统一交给 [DoraemonGPSPMocker sharedInstance]处理 2、[DoraemonGPSPMocker sharedInstance]绑定了所有CLLocationManager与它的delegate的一一对应关系。3、处理 [DoraemonGPSPMocker sharedInstance]没有实现的selector，并且给用户提示。

第二步：当有一个定位回调过来的时候，我们先传给[DoraemonGPSPMocker sharedInstance]，然后[DoraemonGPSPMocker sharedInstance]再转发给它绑定过的所有的delegate。那我们App为例，绑定关系如下：

```
{
    "0x2800a07a0_binder" = "<CLLocationManager: 0x2800a07a0>";
    "0x2800a07a0_delegate" = "<MAMapLocationManager: 0x2800a04d0>";
    "0x2800b59a0_binder" = "<CLLocationManager: 0x2800b59a0>";
    "0x2800b59a0_delegate" = "<KDDriverLocationManager: 0x2829d3bf0>";
}
```

由此可见，我们App的统一定位KDDriverLocationManager和苹果地图的定位MAMapLocationManager都是使用都是CLLocationManager提供的。

具体 DoraemonGPSPMocker这个类如何实现，请参考DorameonKit/Core/Plugin/GPS中的代码。

## 2.4：H5任意门

### H5任意门.png

有的时候Native和H5开发同时开发一个功能，H5依赖native提供入口，而这个时候Native还没有开发好，这个时候H5开发就没法在App上看到效果。再比如，有些H5页面处于的位置比较深入，就像我们代驾司机端，做单流程比较多，有的H5界面需要很繁琐的操作才能展示到App上，不方便我们查看和定位问题。

这个时候我们可以为app做一个简单的浏览器，输入url，使用自带的容器进行跳转。因为每一个app的H5容器基本上都是自定义过得，都会有自己的bridge定制化，所以这个H5容器没有办法使用系统原生的UIWebView或者WKWebView，就只能交给业务方自己去完成。我们在DorameonKit初始化的时候，提供了一个回调让业务方用自己的H5容器去打开这个Url：

```
[[DoraemonManager sharedInstance] addH5DoorBlock:^(NSString *h5Url) {  
    //使用自己的H5容器打开这个链接  
}];
```

这个工具实现比较简单，就不多说了，代码路径在DorameonKit/Core/Plugin/H5.

## 2.5：子线程UI检查

子线程UI\_1.png

---

子线程UI\_2.png

---

### 子线程UI\_3.png

---

在iOS中是不允许在子线程中对UI进行操作和渲染的，不然会造成未知的错误和问题，甚至会导致crash。我们在最近几个版本中发现新增了一些crash，调查原因就是子线程中操作UI导致的。为了对于这种情况可以提早被我们发现，我在DoraemonKit中增加了子线程UI渲染检查查询。

具体事项思路，我们hook住UIView的三个必须在主线程中操作的绘制方法。1、setNeedsLayout 2、setNeedsDisplay 3、setNeedsDisplayInRect:。然后判断他们是不是在子线程中进行操作,如果是在子线程进行操作的话，打印出当前代码调用堆栈，提供给开发进行解决。具体代码如下：

```

@implementation UIView (Doraemon)

+ (void)load{
    [[self class] doraemon_swizzleInstanceMethodWithOriginSel:@selector(setNeedsLayout) swizzledSel:@selector(doraemon_setNeedsLayout)];
    [[self class] doraemon_swizzleInstanceMethodWithOriginSel:@selector(setNeedsDisplay) swizzledSel:@selector(doraemon_setNeedsDisplay)];
    [[self class] doraemon_swizzleInstanceMethodWithOriginSel:@selector(setNeedsDisplayInRect:) swizzledSel:@selector(doraemon_setNeedsDisplayInRect:)
];
}

- (void)doraemon_setNeedsLayout{
    [self doraemon_setNeedsLayout];
    [self uiCheck];
}

- (void)doraemon_setNeedsDisplay{
    [self doraemon_setNeedsDisplay];
    [self uiCheck];
}

- (void)doraemon_setNeedsDisplayInRect:(CGRect)rect{
    [self doraemon_setNeedsDisplayInRect:rect];
    [self uiCheck];
}

- (void)uiCheck{
    if([[DoraemonCacheManager sharedInstance] subThreadUICheckSwitch]){
        if(![NSThread isMainThread]){
            NSString *report = [BSBacktraceLogger bs_backtraceOfCurrentThread];

            NSDictionary *dic = @{
                @"title": [DoraemonUtil dateFormatNow],
                @"content": report
            };

            [[DoraemonSubThreadUICheckManager sharedInstance].checkArray addObject:dic];
        }
    }
}

@end

```

完整代码实现请参考DorameonKit/Core/Plugin/SubThreadUICheck

## 2.6：日志显示

### 日志显示.png

---

这个主要是方便我们查看本地日志，以前我们如果要查看日志，需要自己写代码，访问沙盒导出日志文件，然后再查看。也是比较麻烦的。

DoraemonKit的解决方案是：我们每一次触发日志的时候，都把日志内容显示到界面上，方便我们查看。

如何实现的呢？因为我们这个工具并不是一个通用性的工具，只针对于底层日志库是CocoaLumberjack的情况。稍微讲一下的CocoaLumberjack原理，所有的log都会发给DDLog对象，其运行在自己的一个GCD队列中，之后，DDLog会将log分发给其下注册的一个或者多个Logger中，这一步在多核下面是并发的，效率很高。每一个Logger处理收到的log也是在它们自己的GCD队列下做的，它们询问其下的Formatter，获取Log消息格式，然后根据Logger的逻辑，将log消息分发到不同的地方。系统自带三个Logger处理器，DDTTYLogger，主要将日志发送到Xcode控制台；DDASLLogger，主要讲日志发送到苹果的日志系统Console.app; DDFileLogger，主要将日志发送到文件中保存起来，也是我们开发用到最多的。但是自带的Logger并不满足我们的需求，我们的需求是将日志显示到UI界面中，所以我们需要新建一个类DoraemonLogger，继承于DDAbstractLogger，然后重写logMessage方法，将每一条传过来的日志打印到UI界面中。

log实现.png

---

这个工具参考LumberjackConsole这个开源项目完成，因为刚出iOS11的时候，作者没有适配，所以我们自己拷贝一份代码出来，自己维护了。完整代码实现请参考DorameonKit/WithLogger中。

### 三、总结

写这篇文章主要是为了能够让大家对于DorameonKit进行快速的了解，大家如果有什么好的想法，或者发现我们的这个项目有bug，欢迎大家去github上提Issues或者直接Pull requests，我们会第一时间处理，也希望我们这个工具集合能在大家的一起努力下，做得更加完善。

如果大家觉得我们这个项目还可以的话，点上一颗star吧。


DoraemonKit项目地址：<https://github.com/didi/DoraemonKit>  
(<https://github.com/didi/DoraemonKit>)

### 四、交流群

<https://javer.oss-cn-shanghai.aliyuncs.com/doraemon/github/DoraemonKitQQ.jpeg>

小礼物走一走，来简书关注我

赞赏支持

 IOS (/nb/4936867)

[举报文章](#)    © 著作权归作者所有



景铭巴巴 (/u/c3c893a27097)

写了 26441 字，被 1821 人关注，获得了 1251 个喜欢  
(/u/c3c893a27097)

+ 关注

iOS开发者

喜欢 | 52



更多分享

开发10年  
全记在这本Java进阶宝典了

String源码分析

分布式架构

微服务架构

JVM性能优化

高效DevOps


多线程并发编程


立即修炼


(/p/7dd2ad568a69)


被以下专题收入，发现更多相似内容





 iOS Dev... (/c/3233d1a249ca?utm\_source=desktop&utm\_medium=notes-included-collection)


 iOS开发之常规控件 (/c/58ff7dac7e53?utm\_source=desktop&utm\_medium=notes-included-collection)

 iOS开发资料收集区 (/c/e1d4254ece46?utm\_source=desktop&utm\_medium=notes-included-collection)

 iOS Review (/c/e76c9e115ba1?utm\_source=desktop&utm\_medium=notes-included-collection)

 iOS开发之常见问题 (/c/a982297ff633?utm\_source=desktop&utm\_medium=notes-included-collection)


 iOS 性能 (/c/1e457faaee2c?utm\_source=desktop&utm\_medium=notes-included-collection)

 iOS 应用层 (/c/32010af56e3a?utm\_source=desktop&utm\_medium=notes-included-collection)

展开更多 ∨

## Android - 收藏集 (/p/dad51f6c9c4d?utm\_campaign=maleskine&utm\_c...

Android 自定义View的各种姿势1 Activity的显示之ViewRootImpl详解 Activity的显示之ViewRootImpl初探 Activity的显示之Window和View Android系统的创世之初以及Activity的生命周期 图解Andro...

 passiontim (/u/e946d18f163c?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommen

## Android - 收藏集 (/p/73e36529cf7c?utm\_campaign=maleskine&utm\_...

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金 Cover 有什么料？从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？ ... Android 获取 View 宽高的常用正确方式，避免为零 -

 hw1212 (/u/4031c6d0bd2b?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommen


(/p/d7583ba189d7?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommen  
皮肤管理培训 给你一只杠杆 你可以撬动地球吗 (/p/d7583ba189d7?utm\_...

近些年来，随着爱美人士的增加，皮肤管理逐渐成为人们所关注的焦点，然而国内这个行业的工作人员并

不充裕。如果想要进入到这个行业更好的发展，步就是要到专业的培训机构来让自己得到非常好的本领，

 ASDFS DG (/u/f457ef4634eb?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommen


(/p/0248aaf930ef?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommen

**面试官问“你最想做的工作是什么”，该怎样回答比较合适？ (/p/0248aaf9...**

一次同事面试有这样一个问题，你最想做什么行业，但现在面试的是销售.....这位同事就懵了，怎么合理的回答这个问题？ 面试官询问你的职业发展意向，其实目的很简单：我们公司是不是你的菜，你的职业规划

 明哥聊求职 (/u/5377c30aa323?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommen


(/p/924c4f180384?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommen

**转载|闲叙 (/p/924c4f180384?utm\_campaign=maleskine&utm\_conten...**


连夜拔草几本书 格子说买书的获得的快乐主要来源于大脑对自己一而再再而三的过高期许 深以为然 林奕含的书难读 推进很慢 不知道是因为繁体竖版的排版缘故还是每个字句都可以反复阅读联系前后情节 再联

 一丢丢DeiDeiBei (/u/e9521a3d9313?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommen

**重逢 (/p/e0465d91ebad?utm\_campaign=maleskine&utm\_content=not...**


见到你们之前，我满脑子想的都是一句话“怪风太温柔，像老朋友，像旧时候”。你知道那种心怀期待又心怀紧张的感觉么？ 前天晚上一直在想象会以怎样的方式相逢，比如拥抱，比如我出现在你们身后拍一下你

 声白 (/u/323d4af615bd?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommen

**鼎然金刚藏法语2017.12.19 (/p/7d2c939fd0e6?utm\_campaign=maleski...**

当我们有了太多追求，我们的心就会乱作一团。我们不妨试试追求一种值得让自己喜欢而为之信仰的。你总会发现你所追求的正是你所真正喜欢并为之付出的。那么，你的一生即便是忙碌的也是值得的。人这一

 本乘 (/u/f09bc04c2ca0?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommen

