

详解iOS 轻松获取当前控制器的正确方式

背景

在开发过程中,经常需要获取当前 window, rootViewController, 以及当前正在显示的 visibleController 的需求. 如果 .m 实现不是在当前视图情况下, 我们需要快速的获取到当前控制器, 这种情况就需要先做好一层封装,我一般是通过 UIViewController 写的一个 Category 来实现, 实现起来也非常简单, 只需要我们对 控制器几个方法掌握便可。

获取根控制器

```
+ (UIViewController *)jsd_getRootViewController{

    UIWindow* window = [[[UIApplication sharedApplication] delegate] window];
    NSAssert(window, @"The window is empty");
    return window.rootViewController;
}
```

这里很简单, 通过单例获取到当前 UIApplication 的 delegate 在通过 window 即可轻松拿到 rootViewController。

获取当前页面控制器

```
+ (UIViewController *)jsd_findVisibleViewController {

    UIViewController* currentViewController = [self jsd_rootViewController];

    BOOL runLoopFind = YES;
    while (runLoopFind) {
        if (currentViewController.presentedViewController) {
            currentViewController = currentViewController.presentedViewController;
        } else {
            if ([currentViewController isKindOfClass:[UINavigationController class]]) {
                currentViewController = ((UINavigationController *)currentViewController).visibleViewController;
            } else if ([currentViewController isKindOfClass:[UITabBarController class]]) {
                currentViewController = ((UITabBarController *)currentViewController).selectedViewController;
            } else if ([currentViewController isKindOfClass:[UISplitViewController class]]) { // 当需要兼容 Ipad 时
                currentViewController = currentViewController.presentingViewController;
            } else {
                if (currentViewController.presentingViewController) {
                    currentViewController = currentViewController.presentingViewController;
                } else {
                    return currentViewController;
                }
            }
        }
    }

    return currentViewController;
}
```

这里讲一下实现思路, 我们想要与控制器无耦合的情况下, 想要直接获取到当前控制器, 基本都是通过 rootViewController 来查找的, 通过上面的方法拿到 rootViewController 之后, 我们先看 presentedViewController, 因为控制器呈现出来的方式有 push 与 present, 我们先查看它是否是 present 出来的, 如果是则通过此属性能找到 present 出来的当前控制器, 然后在检查是否属于 UINavigationController 或 UITabBarController, 如果是则通过查找其子控制器里面最顶层或者其正在选择的控制器。最后在判断当前控制器是否有子控制器的情况, 如果有则取其子控制器最顶层, 否则当前控制器就是其本身。

这里主要是查找当前 应用程序基于 UITabBarController 和 UINavigationController 下管理的视图控制器, 如果还有其他控制器则需要添加 if 条件来进行判断。

方法二: 当我们有正在呈现的视图控制器子 View 时, 可通过属性 nextResponder 递归查找

```
+ (nullable UIViewController *)findBelongViewControllerForView:(UIView *)view {
    UIResponder *responder = view;
    while ((responder = [responder nextResponder]))
        if ([responder isKindOfClass: [UIViewController class]]) {
            return (UIViewController *)responder;
        }
    return nil;
}
```

presentedViewController

Apple 文档 presentedViewControlle

通过此方法可以查找到通过 presented 模态方式(显示与隐士) 方式推出的当前控制器。 例如: AViewController -> BViewController 通过模态方式推出. 则使用 AViewController.presentedViewController 能获取到 BViewController。

presentingViewController

Apple 文档

通过此方法可以查找到通过 presented 模态方式(显示与隐士) 方式推出当前控制器的上层控制器。 例如: AViewController -> BViewController 通过模态方式推出. 则使用 BViewController.presentingViewController 能获取到 AViewController。

modalViewController

查看文档发现此方法已在 iOS 6 后被弃用, 官方推荐直接使用 presentedViewController 替代即可.

参考资料与Dome

Dome: 轻松获取当前控制器

UIViewController的presentedViewController , presentingViewController和parentViewController三个属性

您可能感兴趣的文章:iOS开发中导航控制器的基本使用教程iOS如何获取当前View所在控制器的方法详解iOS的UI开发中控制器的创建方法iOS通过多种方式创建控制器iOS如何自定义控制器转场动画push详解iOS多控制器实现带滑动动画iOS开发OC代码中创建Swift编写的视图控制器iOS Swift控制器转场动画示例代码iOS初始化控制器的实现方法总结iOS实现容器视图控制器的方法