

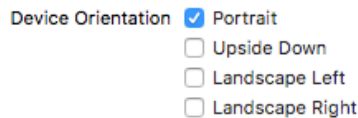
页面旋转和 Frame 纠正

1. 简述:

本文要介绍的是实际项目开发中比较常见的一种场景: App 应用全局仅支持竖屏, 个别页面, 如视频播放页等等, 需要对页面进行旋转。如何实现这种效果呢?

2. 页面旋转实现

- (1) 首先, 有个概念, 应用全局设置: xcode 项目工程中有可视化配置 app 页面旋转支持的地方, 如下



这种设置是应用全局的旋转支持设置项, 还有另一种页面全局支持设置项, 是在代码中设置的

```
-(UIInterfaceOrientationMask)application:(UIApplication *)application supportedInterfaceOrientationsForWindow:(UIWindow *)window {
    return UIInterfaceOrientationMaskAll;
}
```

当两种应用全局都设置的情况下, 以代码中的设置为主

- (2) 其次, 另一个概念: vc 级设置, viewController 有一些对屏幕旋转支持设置的方法, 如下:

```
// !!! 处理视频全屏展示
-(BOOL)shouldAutorotate{
    //允许自动旋转
    return YES;
}
-(UIInterfaceOrientationMask)supportedInterfaceOrientations{
    //支持所有方向的旋转
    return UIInterfaceOrientationMaskAll;
}
-(UIInterfaceOrientation)preferredInterfaceOrientationForPresentation{
    //页面默认旋转方向
    return UIInterfaceOrientationPortrait;
}
```

注意: 如果你希望某个页面 preferd 是横屏, 那么 support 里返回的必须有横屏项, 不然会抛出异常。

- (3) 具体实现:

1) 代码中

```

-(UIInterfaceOrientationMask)application:(UIApplication *)application supportedInterfaceOrientationsForWindow:(UIWindow *)window
if (_allowRotate == 1) {
    //    return UIInterfaceOrientationMaskAll;

```

具体需要旋转的页面，实现 vc 级别的三个方法进行设置

在旋转的时候将代码中的_allowRotate 设为 1(全局设置支持全部)

旋转结束的时候将其置回 0(全局设置仅竖屏)

2) 使用 ViewController 的分类

分类中实现

```

@implementation UIViewController (Orientation)
// 所支持的方向运行旋转
-(BOOL)shouldAutorotate{
    if ([[self class] isKindOfClass:[ShiPinBoFangViewController class]]) {
        return NO;
    }
    return YES;
}
// 默认VC竖屏
-(UIInterfaceOrientationMask)supportedInterfaceOrientations
{
    if ([UIApplication sharedApplication].keyWindow == [UIApplication sharedApplication].windows[0]) { appDelegate的window是第0个 }
    if ([[self class] isKindOfClass:[KLineshupViewController class]]) {
        return UIInterfaceOrientationMaskLandscape;
    }
    return UIInterfaceOrientationMaskPortrait;
} else {
    return UIInterfaceOrientationMaskAllButUpsideDown;
}
}
// 默认的屏幕方向（当前ViewController必须是通过模态出来的UIViewController（模态带导航的无效）方式展现出来的，才会调用这个方法）
-(UIInterfaceOrientation)preferredInterfaceOrientationForPresentation{
    if ([[self class] isKindOfClass:[KLineshupViewController class]]) {
        return UIInterfaceOrientationLandscapeRight;
    }
    return UIInterfaceOrientationPortrait;
}
@end

```

删除所有系统级设置，包括删除代码设置、取消勾选 xcode 配置文件中的相关设置

分类实现：默认支持旋转、支持方向默认竖屏、当页面为指定页面时候(如自定义的全屏页面)支持横屏、当页面为点击网页里的视频播放按钮弹出的页面时候支持竖、横屏

3) 经测试，以上方法均有一些问题，接下来介绍的方法是目前使用觉得还 ok 的方法：

同样，删除所有系统级设置，包括删除代码设置、取消勾选 xcode 配置文件中的相关设置

在根页面如 tabBarController 重写三个旋转方法

```

// 所支持的方向运行旋转
-(BOOL)shouldAutorotate{

    return NO;
}
// 默认VC竖屏
-(UIInterfaceOrientationMask)supportedInterfaceOrientations
{
    return UIInterfaceOrientationMaskPortrait;
}
-(UIInterfaceOrientation)preferredInterfaceOrientationForPresentation{
    return UIInterfaceOrientationPortrait;
}

```

这样，默认就都是竖屏了，哪个页面想支持循环的话，只要重写下这三个方法即可

(4) 页面旋转的 Frame 纠正

页面旋转横屏与竖屏的切换，有一个很头疼的问题，那就是屏幕的宽高变了，width 可能变为 height、height 也可能变为 width。如果是 autolayout 布局的，在旋转的时候，系统内部会自动帮我们重新设置 frame。那如果我们页面采用的是 frame 布局的话，就需要我们自己做这一项工作了。具体如下：

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(controllerIsRotae) name:UIApplicationDidChangeStatusBarOrientationNotification object:nil];
}

#pragma mark - <***** 事件处理 *****>

- (void)controllerIsRotae{
    self.contentHeight.constant = [self.contentLabel sizeThatFits:CGSizeMake(IEW_WIDTH - 24, CGFLOAT_MAX)].height;
    self.viewHeight = self.contentHeight.constant + 48 + 45;
    [self layoutIfNeeded];
}
```

通过注册通知，监听屏幕即将旋转或者屏幕完成旋转事件，然后进行相应的处理