

## Getting the Device Orientation with UIKit

When you need to know the **general orientation of a device, and not its precise orientation**, get that information from the shared `UIDevice` object. Device orientation indicates whether the device is upright in portrait or landscape orientation or whether the device is face up or face down. For some apps, this might be enough information to make decisions about how to present content.

Before you can retrieve orientation-related data from the `UIDevice` object, you must call that object's `beginGeneratingDeviceOrientationNotifications` method. Calling that method enables the accelerometer hardware, which is needed to determine the device's orientation. UIKit updates the `orientation` property of `UIDevice` and delivers `UIDeviceOrientationDidChange` notifications to any registered observers of your app when the orientation value changes. **When you no longer need the orientation information, call the `endGeneratingDeviceOrientationNotifications` method. Doing so gives the system the option to save power by turning off the accelerometer hardware.**

Listing 14-1 shows how to use the `viewWillAppear(_:)` and `viewWillDisappear(_:)` methods of your view controller to enable and disable the delivery of orientation notifications to your app. The notification does not contain the orientation value, but you can always get the current value from the `orientation` property.

**Listing 14-1** Responding to changes in device orientation

```

OBJECTIVE-C
1  - (void)viewWillAppear:(BOOL)animated {
2      [super viewWillAppear:animated];
3
4      UIDevice* device = [UIDevice currentDevice];
5      [device beginGeneratingDeviceOrientationNotifications];
6
7      // Register for orientation change notifications.
8      [[NSNotificationCenter defaultCenter] addObserver:self
9          selector:@selector(updateOrientation:)
10         name:UIDeviceOrientationDidChangeNotification
11         object:nil];
12 }
13
14 - (void)viewWillDisappear:(BOOL)animated {
15     [super viewWillDisappear:animated];
16
17     UIDevice* device = [UIDevice currentDevice];
18     [device endGeneratingDeviceOrientationNotifications];
19
20     // Clean up.
21     [[NSNotificationCenter defaultCenter] removeObserver:self];
22 }
23
24 - (void)updateOrientation:(NSNotification*) notification {
25     UIDeviceOrientation orientation = [UIDevice currentDevice].orientation;
26     // Use the orientation value in your app.
27 }

SWIFT
1  override func viewWillAppear(_ animated: Bool) {
2      super.viewWillAppear(animated)
3
4      UIDevice.current.beginGeneratingDeviceOrientationNotifications()
5      self.observer = NotificationCenter.default.addObserver(
6          forName: .UIDeviceOrientationDidChange,
7          object: nil, queue: self.queue) { (notification) in
8              let orientation = UIDevice.current.orientation
9              // Use the orientation value in your app.
10         }
11 }

```

```
12  
13  override func viewDidLoadDisappeared(_ animated: Bool) {  
14      super.viewDidLoadDisappeared(animated)  
15  
16      UIDevice.current.endGeneratingDeviceOrientationNotifications()  
17      NotificationCenter.default.removeObserver(self.observer!)  
18  }
```

Copyright © 2017 Apple Inc. All rights reserved. [Terms of Use](#) | [Privacy Policy](#) | [Updated: 2017-03-21](#)