

PERMISSION

Zeke Abuhoff

Lead iOS Instructor, General Assembly

PERMISSION

LEARNING OBJECTIVES

- + Explain iOS' permission system
- + Request permission for photo access
- + Request permission for location access
- + Handle a lack of permission
- + Acquire familiarity with other iOS permission processes

PERMISSION

PERMISSION

In iOS, the user can grant an app permission to access sensitive data, like the user's photos or the user's location.

How does granting permission work?

PERMISSION

PHOTO PERMISSION

The Photos framework allows us to access the user's photo library, but we need the user's permission.

The `requestAuthorization` method takes a closure in which you react to each possible authorization state.

```
PHPhotoLibrary.requestAuthorization {
    status in
        switch status {
        case .authorized:
            print("Authorized")
        case .restricted:
            print("Restricted")
        case .denied:
            print("Denied")
        default:
            print("DEFAULT")
        }
}
```

PERMISSION

PHOTO PERMISSION

But this code alone won't work!

The Photos framework also requires

`NSPhotoLibraryUsageDescription` as a key in your Info.plist file. This key requires a value.

The value for this key is text displayed when the permission dialogue appears.

```
PHPhotoLibrary.requestAuthorization {
    status in
        switch status {
        case .authorized:
            print("Authorized")
        case .restricted:
            print("Restricted")
        case .denied:
            print("Denied")
        default:
            print("DEFAULT")
        }
}
```

PERMISSION

LOCATION PERMISSION

Location tracking requires user permission also.

Permission can be acquired for either gathering location data when the app is in use or gathering location data at any time, even if the app is in the background.

The code to request permission is so short!

```
CLLocationManager().requestWhenInUseAuthorization()
```

or

```
CLLocationManager().requestAlwaysAuthorization()
```

PERMISSION

LOCATION PERMISSION

But this code won't work on its own either!

We need

`NSLocationWhenInUseUsageDescription` or

`NSLocationAlwaysUsageDescription` in the Info.plist with corresponding values.

```
CLLocationManager().requestWhenInUseAuthorization()
```

or

```
CLLocationManager().requestAlwaysAuthorization()
```

PERMISSION

LOCATION PERMISSION

We also need to make sure that the location manager object we create stays in memory.

When a location manager is discarded because its scope ends, its permission requests are cut short.

We can solve this problem by defining the location manager at the level of an instance property.

```
let locationManager =  
CLLocationManager()
```

```
locationManager.requestWhenInUse  
Authorization()
```

or

```
locationManager.requestAlwaysAut  
horization()
```

PERMISSION

LACK OF PERMISSION

Of course, the user can always answer our requests for permission with “no.”

Once the user denies permission, your app cannot ask again. The permission status for that feature can only change if the user visits their own settings and flips a switch themselves.

Users never do this.

Once your user has refused you permission, you should assume you’re never getting permission and try to provide the best possible experience within that constraint.

PERMISSION

LACK OF PERMISSION



Because the stakes for a permission request are so high, don't make the request too early. If you surprise the user with a permission request, your app comes off as sketchy and invasive. Instead, wait until the user can see why you need that particular permission and preface the system-level request with one you made yourself - one that you can ask again later if you want.