Article

Authorizing access to health data

Request permission to read and share data in your app.

Overview

To help protect people’s privacy, HealthKit requires fine-grained authorization. You need to request permission to both read and share each data type before your app attempts to use the data. However, you don’t need to request permission for all data types at once. Instead, it might make more sense to wait until you need to access the data before asking for permission.

As part of the privacy protections, your app doesn’t know whether someone granted or denied permission to read data from HealthKit. If they denied permission, attempts to read data from HealthKit return only samples that your app successfully saved to the HealthKit store. Additionally, in a Guest User session on Apple Vision Pro, the guest can view previously authorized data, but can’t access unauthorized data or change the authorizations.

Important

In iOS 17.2 and later, the Journal app encourages people to reflect on their day-to-day experiences, including physical accomplishments, workouts, and emotions and moods. If your app saves data to HealthKit, high-level summaries of that data can appear as suggestions in the Journal app, or in other apps that use the Journaling Suggestions framework.

Requesting permission to read and share data is only one part of protecting your user’s privacy. For more information, see Protecting user privacy.

Enable HealthKit

Before you can request authorization to read or save HealthKit data, you need to add the HealthKit capability to your app. You must also provide custom messages for the Health permissions sheet.

Xcode requires separate custom messages for reading and writing HealthKit data. Set the `NSHealthShareUsageDescription` key to customize the message for reading data and the `NSHealthUpdateUsageDescription` key to customize the message for writing data.

For projects created using Xcode 13 or later, set these keys in the Target Properties list on the app's Info tab. For projects created with Xcode 12 or earlier, set these keys in the app's `Info.plist` file. For more information, see `Information Property List`.

Finally, check that Health data is available on the current device by calling `isHealthDataAvailable()` before calling any other HealthKit methods. For more information, see Setting up HealthKit.

Request permission

To request permission to read or write data, start by creating the HealthKit data types that you want to read or write. The following example creates data types for active energy burned, distance cycling, distance walking or running, distance in a wheelchair, and heart rate.

```
// Create the HealthKit data types your app
// needs to read and write.
let allTypes: Set = [
    HKQuantityType.workoutType(),
    HKQuantityType(.activeEnergyBurned),
    HKQuantityType(.distanceCycling),
    HKQuantityType(.distanceWalkingRunning),
    HKQuantityType(.distanceWheelchair),
    HKQuantityType(.heartRate)
]
```

Next, you can request read or write access to that data. To request access from the HealthKit store, call `requestAuthorization(toShare:read:)`.

```
do {
    // Check that Health data is available on the device.
    if HKHealthStore.isHealthDataAvailable() {

        // Asynchronously request authorization to the data.
        try await healthStore.requestAuthorization(toShare: allTypes, read: allTypes
    }
} catch {

    // Typically, authorization requests only fail if you haven't set the
    // usage and share descriptions in your app's Info.plist, or if
    // Health data isn't available on the current device.
    fatalError("*** An unexpected error occurred while requesting authorization: \(e
}
```

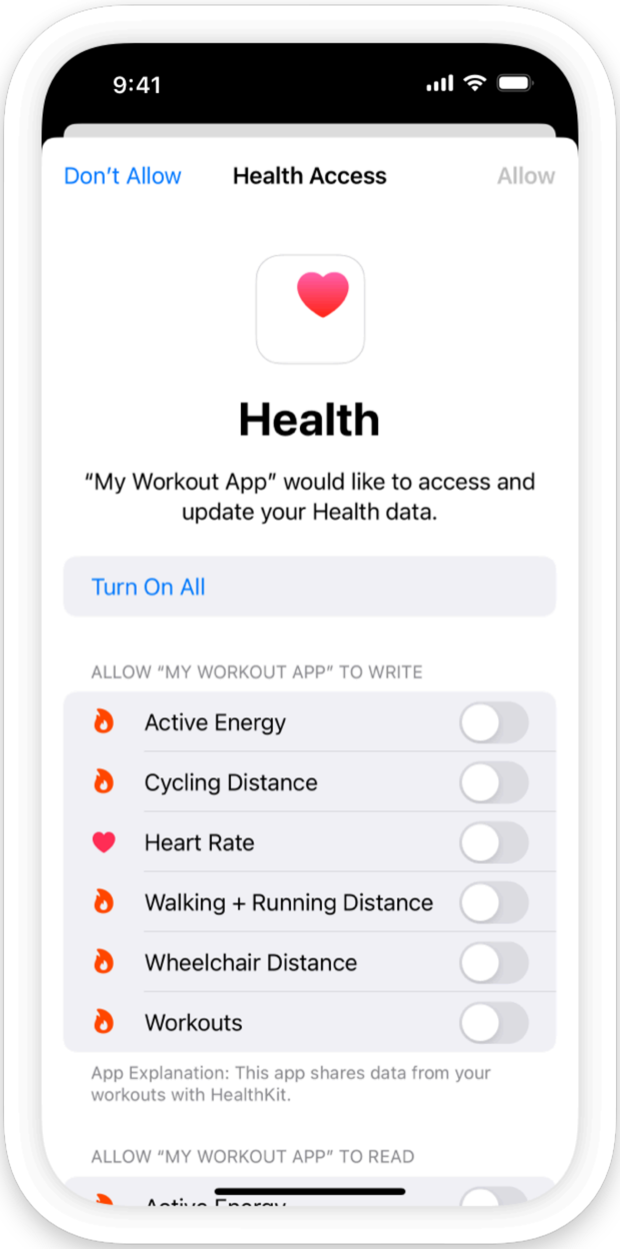
To request access from SwiftUI, use the healthDataAccessRequest(store:shareTypes:readTypes:trigger:completion:) modifier.

Important

The healthDataAccessRequest(store:shareTypes:readTypes:trigger:completion:) modifier is only available if you import both SwiftUI and HealthKitUI.

```
import SwiftUI
import HealthKitUI

struct MyView: View {
    @State var authenticated = false
    @State var trigger = false

    var body: some View {
        Button("Access health data") {
            // OK to read or write HealthKit data here.
        }
        .disabled(!authenticated)

        // If HealthKit data is available, request authorization
        // when this view appears.
        .onAppear() {

            // Check that Health data is available on the device.
            if HKHealthStore.isHealthDataAvailable() {
                // Modifying the trigger initiates the health data
                // access request.
                trigger.toggle()
            }
        }

        // Requests access to share and read HealthKit data types
        // when the trigger changes.
        .healthDataAccessRequest(store: healthStore,
                                 shareTypes: allTypes,
                                 readTypes: allTypes,
                                 trigger: trigger) { result in
            switch result {

            case .success(_):
                authenticated = true
            case .failure(let error):
                // Handle the error here.
                fatalError("*** An error occurred while requesting authentication: \
            }
        }
    }
}
```

Any time your app requests new permissions, the system displays a form with all the requested data types shown. People can toggle individual read and share permissions on and off.



9:41

Don't Allow | Health Access | Allow

Health

“My Workout App” would like to access and update your Health data.

Turn On All

ALLOW “MY WORKOUT APP” TO WRITE

- Active Energy ○
- Cycling Distance ○
- Heart Rate ○
- Walking + Running Distance ○
- Wheelchair Distance ○
- Workouts ○

App Explanation: This app shares data from your workouts with HealthKit.

ALLOW “MY WORKOUT APP” TO READ

- Active Energy ○

To learn how to provide a great experience when asking for permissions, see Human Interface Guidelines > HealthKit.

> **Important**
>
> People can change the permissions for your app at any time using either the Settings or the Health app. Your app appears in the Health app's Sources tab, even if they didn't allow permission to read or share data.

Check for authorization before saving data

If someone grants permission to share a data type, you can create new samples of that type and save them to the HealthKit store. However, before attempting to save any data, check to see if your app is authorized to share that data type by calling the `authorizationStatus(for:)` method. If you haven't yet requested permission, any attempts to save fail with an `HKError.Code.errorAuthorizationNotDetermined` error. If they've denied permission, attempts to save fail with an `HKError.Code.errorAuthorizationDenied` error.

Support Guest User sessions on Vision Pro

To protect their privacy, people can put their Vision Pro in a Guest User session before sharing it. This session lets the owner control which apps the guest can use, and what data they can see. For more information, refer to Let another person use your Apple Vision Pro with Guest User.

A Guest User session has the following affects on HealthKit:

- If the owner has already authorized access to the data, the guest can read that data from the HealthKit store.
- The guest can't authorize any additional data types.
- The system obscures Health data in the Privacy and Security and Health Data panels in Settings.
- Any attempts to save data or otherwise mutate data in the HealthKit store fails with an `HKError.Code.errorNotPermissibleForGuestUserMode` error (or `HKError.Code.errorHealthDataRestricted` on apps running in iOS 17).

Important

An app’s permissions don’t change when an app runs in a Guest User session. Therefore, `authorizationStatus(for:)` returns `true` if the owner previously granted authorization to write the data, even though the app can’t write it during a Guest User session.

Any attempt to request authorization for HealthKit data types fails silently. The system doesn’t display the authorization sheet during a Guest User session.

If your app receives an `HKError.Code.errorNotPermissibleForGuestUserMode` error, you can silently ignore the error for passive or periodic saves. Silently dropping the changes ensures that they don’t persist past the Guest User session without interrupting the guest’s experience. However, if the guest performs an action that would obviously result in saving data (for example, tapping a Save button), you can display an alert telling them that the action isn’t available during a Guest User session.

Specify required clinical record types

If your app requires access to specific clinical record data to function properly, specify the required clinical record types in your app’s `Info.plist` file using the `NSHealthRequiredReadAuthorizationTypeIdentifiers` key. This key defines the data types that your app must have permission to read. Set the value to an array of strings containing the type identifiers for your required types. For a list of type identifiers, see `HKClinicalTypeIdentifier`.

To protect people’s privacy, you must specify three or more required clinical record types. If a person denies authorization to any of the types, authorization fails with an `HKError.Code.errorRequiredAuthorizationDenied` error; the system doesn’t tell your app which record types the person denied access to.

See Also

Essentials

About the HealthKit framework

Learn about the architecture and design of the HealthKit framework.

Setting up HealthKit

Set up and configure your HealthKit store.

Protecting user privacy

Respect and safeguard your user’s privacy.

HealthKit updates

Learn about important changes to HealthKit.

HealthKitUI

Display user interface that enables a person to view and interact with their health data.