

Framework

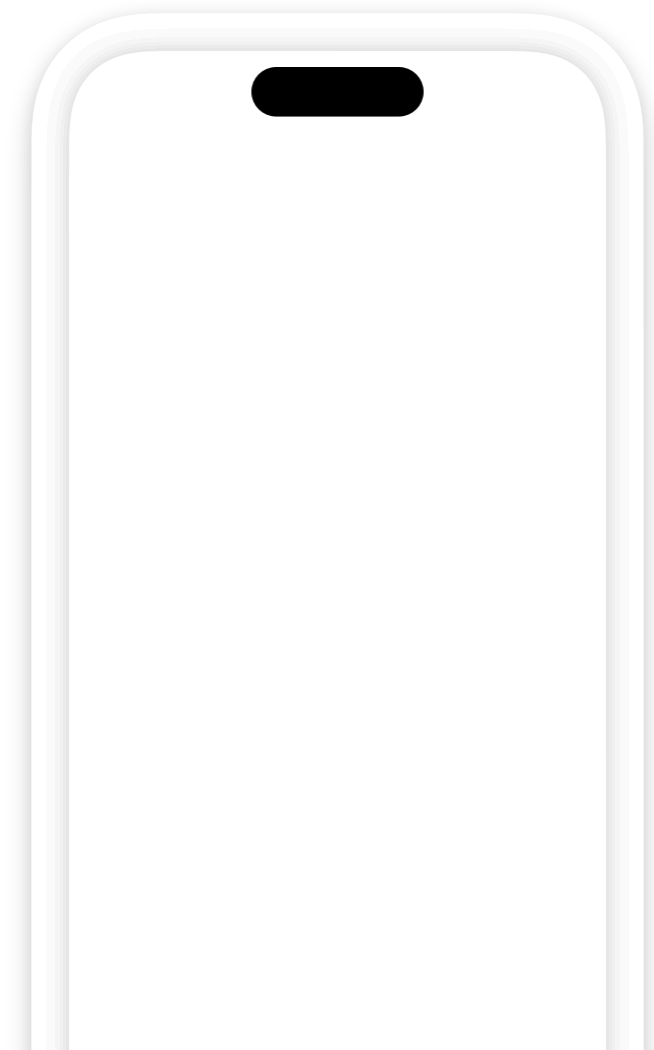
HealthKit

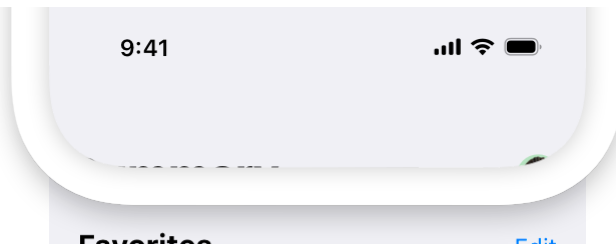
Access and share health and fitness data while maintaining the user's privacy and control.

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 17.0+ | macOS 14.0+ | visionOS 1.0+ | watchOS 2.0+

Overview

HealthKit provides a central repository for health and fitness data on iPhone and Apple Watch. With the user's permission, apps communicate with the HealthKit store to access and share this data.





Creating a complete, personalized health and fitness experience includes a variety of tasks:

- Collecting and storing health and fitness data
- Analyzing and visualizing the data
- Enabling social interactions

HealthKit apps take a collaborative approach to building this experience. Your app doesn't need to provide all of these features. Instead, you can focus just on the subset of tasks that most interests you.

For example, users can select their favorite weight-tracking, step-counting, and health challenge app, each calibrated to their personal needs. Because HealthKit apps freely exchange data (with user permission), the combined suite provides a more customized experience than any single app on its own. For example, when a group of friends joins a daily step-counting challenge, each person can use their preferred hardware device and app to track their steps, while everyone in the group uses the same social app for the challenge.

HealthKit is also designed to manage and merge data from multiple sources. For example, users can view and manage all of their data in the Health App, including adding data, deleting data, and changing an app's permissions. Therefore, your app needs to handle these changes, even when they occur outside your app.

Note

Because health data may contain sensitive, personal information, apps must receive permission from the user to read data from or write data to the HealthKit store. They must also take steps to protect that data at all times. For more information, see [Protecting user privacy](#).






Topics

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.
-  **Authorizing access to health data**
Request permission to read and share data in your app.
-  **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.

Health data

-  **Saving data to HealthKit**
Create and share HealthKit samples.
-  **Reading data from HealthKit**
Use queries to request sample data from HealthKit.
- `class HKHealthStore`
The access point for all data managed by HealthKit.
-  **Creating a Mobility Health App**
Create a health app that allows a clinical care team to send and receive mobility data.
-  **Data types**
Specify the kind of data used in HealthKit.
-  **Samples**
Create and save health and fitness samples.
-  **Queries**
Query health and fitness data.
-  **Visualizing HealthKit State of Mind in visionOS**
Incorporate HealthKit State of Mind into your app and visualize the data in visionOS.
-  **Logging symptoms associated with a medication**

Fetch medications and dose events from the HealthKit store, and create symptom samples to associate with them.

Workout data

☰ Workouts and activity rings

Manage workouts, workout sessions, and activity summaries.

Errors

```
struct HKError
```

An error returned from a HealthKit method.

```
let HKErrorDomain: String
```

The domain for all HealthKit errors.

```
enum Code
```

Error codes returned by HealthKit.

Reference

☰ HealthKit Enumerations

☰ HealthKit Classes

☰ HealthKit Constants

☰ HealthKit Data Types

☰ HealthKit Functions

☰ Macros

☰ HealthKit Variables