

[HealthKit](#) / Saving data to HealthKit

Article

Saving data to HealthKit

Create and share HealthKit samples.

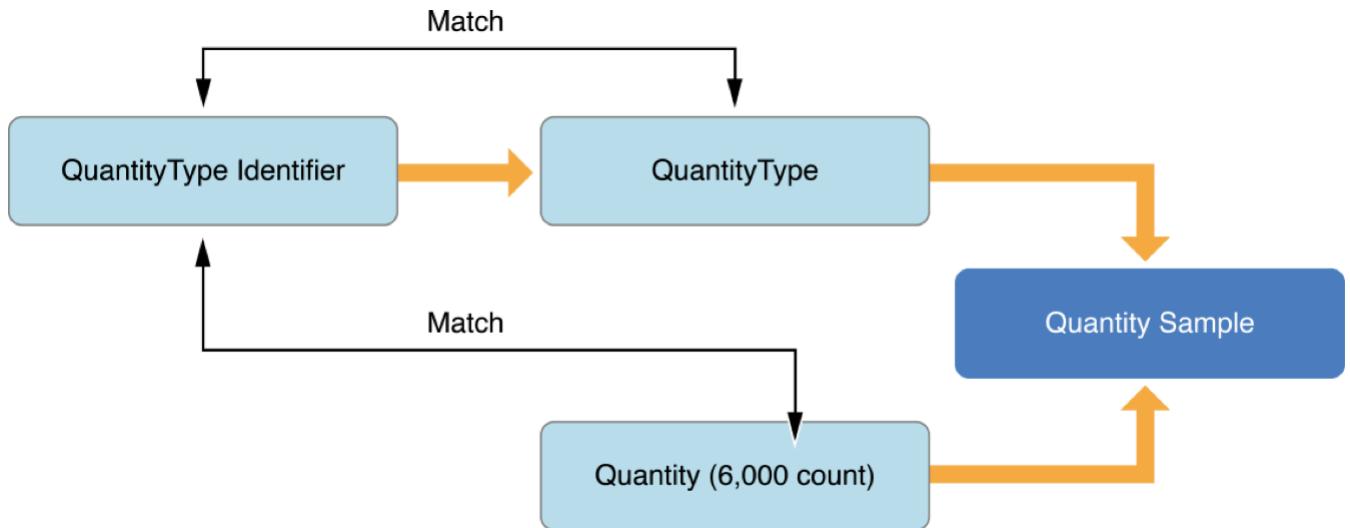


Overview

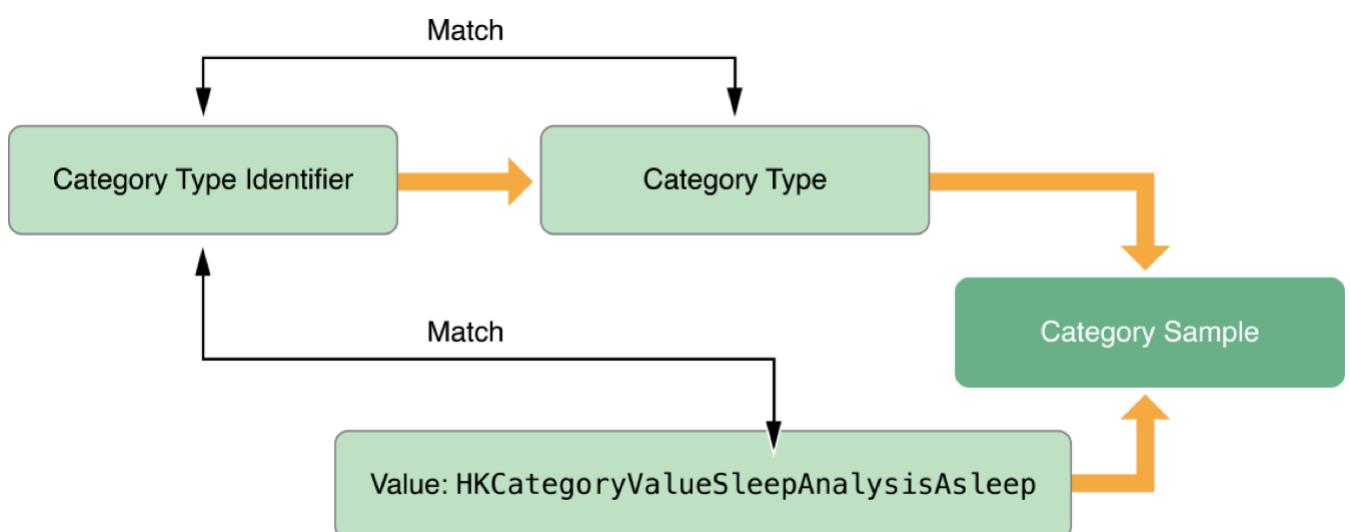
Your app can create new samples and add them to the HealthKit store. Although all sample types follow a similar procedure, each type has its own variations:

1. Look up the type identifier for your data. For example, to record the user's weight, you use the [bodyMass](#) type identifier. For the complete list of type identifiers, see [Data types](#).
2. Use the convenience methods on the [HKObjectType](#) class to create the correct object type for your data. For example, to save the user's weight, you'd create an [HKQuantityType](#) object using the [quantityType\(forIdentifier:\)](#) method. For a list of convenience methods, see [HKObjectType](#).
3. Instantiate an object of the matching [HKSamp_le](#) subclass using the object type.
4. Save the object to the HealthKit store using the [save\(:withCompletion:\)](#) method.

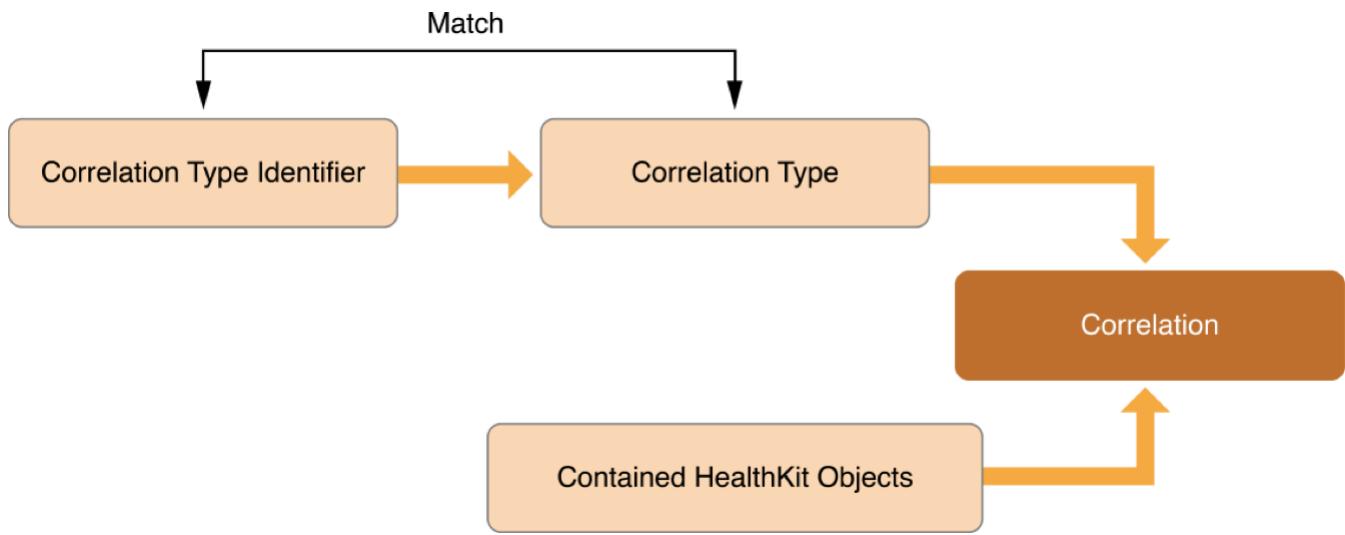
Each [HKSamp_le](#) subclass has its own convenience methods for instantiating sample objects, which modify the steps described in the list above.



For quantity samples, create an instance of the [HKQuantity](#) class. The quantity's units must correspond to the allowable units described in the type identifier's documentation. For example, the [height](#) documentation states that it uses length units. Therefore, your quantity must use centimeters, meters, feet, inches, or another compatible unit. For more information, see [HKQuantitySample](#).



For category samples, the sample's value must correspond to the enum described in the type identifier's documentation. For example, the [sleepAnalysis](#) documentation states that it uses the [HKCategoryValueSleepAnalysis](#) enum. Therefore, you must pass a value from this enum when creating this sample. For more information, see [HKCategorySample](#).



For correlations, you must first create all the sample objects that the correlation will contain. The correlation's type identifier describes both the type and the number of objects that can be contained. Don't save the contained objects into the HealthKit store. They're stored as part of the correlation. For more information, see [HKCorrelation](#).

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.

Balance performance and details

When saving data to the HealthKit store, you often need to choose between using a single sample to represent the data or splitting the data across multiple, smaller samples. A single, long sample is better from a performance perspective; however, multiple smaller samples gives the user a more detailed look into how their data changes over time. Ideally, you want to find a sample size that's granular enough to provide the user with useful historical data and you should avoid saving samples that are 24 hours long or longer.

When recording a workout, you can use high frequency data (a minute or less per sample) to provide intensity charts and otherwise analyze the user's performance over the workout. For less intensive activity, like daily step counts, samples of an hour or less often work best. This lets you produce meaningful daily and hourly graphs.

Most sample types have restrictions on duration. If you attempt to save a sample that doesn't meet those restrictions, it fails to save. For more details on checking the duration restrictions, refer to [HKSampleType](#).

Work with data in the Health app

The Health app gives users access to all of the data in their HealthKit store. Users can view, add, delete, and manage their data.

Specifically, users can:

- See a dashboard containing their current health data.
- Access all the data stored in HealthKit. Users can view the data by type, by app, or by device.
- Manage each app's permission to read and write from the HealthKit store.

As a result, the Health app has a few important impacts on developing HealthKit apps. First, remember that users can always modify their data outside your app or even change your permission to access their data. As a result, your app should always query for the current data in the HealthKit store or perform background queries to track changes to the store.

Second, you can also use the Health app to view the data your app is saving to the HealthKit store. This can be particularly useful during early testing, to verify that your app is saving everything as expected.

See Also

Health data

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.