Article

Reading data from HealthKit

Use queries to request sample data from HealthKit.

Overview

There are three main ways to access data from the HealthKit Store:

- **Direct method calls.** The HealthKit store provides methods to directly access characteristic data. These methods only access characteristic data. For more information, see `HKHealthStore`.
- **Queries.** Queries return the current snapshot of the requested data from the HealthKit store.
- **Long-running queries.** These queries continue to run in the background and update your app whenever the system detects changes to the HealthKit store.

Queries

Queries return the current snapshot of the data in the HealthKit store. All queries run on an anonymous background queue. When the query is complete, it executes the results handler on the background queue. HealthKit provides different types of queries, each designed to return different types of data from the HealthKit store.

- **Sample query.** This is a general-purpose query. Use sample queries to access any type of sample data. Sample queries are particularly useful when you want to sort the results or limit the total number of samples returned. For more information, see `HKSampleQueryDescriptor`.
- **Anchored object query.** Use this query to search for changes to the HealthKit store. The first time you run an anchor query, it returns all the matching samples currently in the store. On subsequent runs, it returns only those items added or deleted since the last run. For more information, see `HKAnchoredObjectQueryDescriptor`.

- **Statistics query.** Use this query to perform statistical calculations over the set of matching samples. You can use statistics queries to calculate the sum, minimum, maximum, or average value in the set. For more information, see `HKStatisticsQueryDescriptor`.
- **Statistics collection query.** Use this query to perform multiple statistics queries over a series of fixed-length time intervals. You often use these queries when creating graphs. They provide a simple method for calculating things such as the total number of calories consumed each day or the number of steps taken during each five-minute interval. For more information, see `HKStatisticsCollectionQueryDescriptor`.
- **Correlation query.** Use this query to perform complex searches of the data contained in a correlation. These queries can contain individual predicates for each of the sample types stored in the correlation. If you just want to match the correlation type, use a sample query instead. For more information, see `HKCorrelationQuery`.
- **Source query.** Use this query to search for sources (apps and devices) that have saved matching samples to the HealthKit store. A source query lists all the sources that are saving a particular sample type. For more information, see `HKSourceQueryDescriptor`.
- **Activity summary query.** Use this query to search for activity summary information for the user. Each activity summary object contains a summary of the user's activity for a given day. You can query for either a single day or a range of days. For more information, see `HKActivitySummaryQueryDescriptor`.
- **Document query.** Use this query to search for health documents. For more information, see `HKDocumentQuery`.

Long-running queries

Long-running queries continue to run an anonymous background queue, and update your app whenever the system detects changes to the HealthKit store. In addition, observer queries can register for background delivery. This lets HealthKit wake your app in the background whenever an update occurs.

HealthKit provides the following long-running queries:

- **Observer query.** This long-running query monitors the HealthKit store and alerts you to any changes to matching samples. Use an observer query when you want the system to notify you about changes to the store. You can register observer queries for background delivery. For more information, see `HKObserverQuery`.
- **Anchored object query.** In addition to returning the current snapshot of modified data, an anchored object query can act as a long-running query. If enabled, it continues to run in the background, providing updates when something adds or removes matching samples from the store. Unlike the observer query, these updates include a list of items that have changed;

however, you can't register anchored object queries for background delivery. For more information, see `HKAnchoredObjectQueryDescriptor`

- **Statistics collection query.** In addition to calculating the current snapshot of statistical collections, this query can act as a long-running query. If something adds or removes matching samples from the store, this query recalculates the statistics collections and updates your app. You can't register statistics collection queries for background delivery. For more information, see `HKStatisticsCollectionQueryDescriptor`.
- **Activity summary query.** In addition to calculating the current snapshot of the user's activity summary, this query can act as a long-running query. If the user's activity summary data changes, this query recalculates the activity summary and updates your app. You can't register activity summary queries for background delivery. For more information, see `HKActivitySummaryQueryDescriptor`.

See Also

Health data

Saving data to HealthKit

Create and share HealthKit samples.

`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.