

[HealthKit](#) / Data types

API Collection

# Data types

Specify the kind of data used in HealthKit.

## Overview

HealthKit uses [HKObjectType](#) subclasses to identify the different types of data stored in HealthKit, from inherent data that doesn't typically change over time to complex data types that combine multiple samples or compute values over sets of samples.

To create a type object, call the appropriate [HKObjectType](#) class method, and pass in the desired type identifier.

```
let bloodType = HKObjectType.characteristicType(forIdentifier: .bloodType)

let caloriesConsumed = HKObjectType.quantityType(forIdentifier: .dietaryEnergyConsumption)

let sleepAnalysis = HKObjectType.categoryType(forIdentifier: .sleepAnalysis)
```

You can use the resulting object types to request permission to access the data, save new data to the HealthKit store, or read data from the HealthKit store.

## Topics

### Object type subclasses

`class` `HKCharacteristicType`

A type that represents data that doesn't typically change over time.

`class HKQuantityType`

A type that identifies samples that store numerical values.

`class HKCategoryType`

A type that identifies samples that contain a value from a small set of possible values.

`class HKCorrelationType`

A type that identifies samples that group multiple subsamples.

`class HKActivitySummaryType`

A type that identifies activity summary objects.

`class HKAudiogramSampleType`

A type that identifies samples that contain audiogram data.

`class HKElectrocardiogramType`

A type that identifies samples containing electrocardiogram data.

`class HKSeriesType`

A type that indicates the data stored in a series sample.

`class HKClinicalType`

A type that identifies samples that contain clinical record data.

`class HKWorkoutType`

A type that identifies samples that store information about a workout.

`class HKObjectType`

An abstract superclass with subclasses that identify a specific type of data for the HealthKit store.

`class HKSampleType`

An abstract superclass for all classes that identify a specific type of sample when working with the HealthKit store.

## Characteristic identifiers

`static let activityMoveMode: HKCharacteristicTypeIdentifier`

A characteristic identifier for the user's activity mode.

`static let biologicalSex: HKCharacteristicTypeIdentifier`

A characteristic type identifier for the user's sex.

`static let bloodType: HKCharacteristicTypeIdentifier`

A characteristic type identifier for the user's blood type.

`static let dateOfBirth: HKCharacteristicTypeIdentifier`

A characteristic type identifier for the user's date of birth.

`static let fitzpatrickSkinType: HKCharacteristicTypeIdentifier`

A characteristic type identifier for the user's skin type.

`static let wheelchairUse: HKCharacteristicTypeIdentifier`

A characteristic identifier for the user's use of a wheelchair.

## Activity

`static let stepCount: HKQuantityTypeIdentifier`

A quantity sample type that measures the number of steps the user has taken.

`static let distanceWalkingRunning: HKQuantityTypeIdentifier`

A quantity sample type that measures the distance the user has moved by walking or running.

`static let runningSpeed: HKQuantityTypeIdentifier`

A quantity sample type that measures the runner's speed.

`static let runningStrideLength: HKQuantityTypeIdentifier`

A quantity sample type that measures the distance covered by a single step while running.

`static let runningPower: HKQuantityTypeIdentifier`

A quantity sample type that measures the rate of work required for the runner to maintain their speed.

`static let runningGroundContactTime: HKQuantityTypeIdentifier`

A quantity sample type that measures the amount of time the runner's foot is in contact with the ground while running.

`static let runningVerticalOscillation: HKQuantityTypeIdentifier`

A quantity sample type measuring pelvis vertical range of motion during a single running stride.

`static let distanceCycling: HKQuantityTypeIdentifier`

A quantity sample type that measures the distance the user has moved by cycling.

`static let pushCount: HKQuantityTypeIdentifier`

A quantity sample type that measures the number of pushes that the user has performed while using a wheelchair.

```
static let distanceWheelchair: HKQuantityTypeIdentifier
```

A quantity sample type that measures the distance the user has moved using a wheelchair.

```
static let swimmingStrokeCount: HKQuantityTypeIdentifier
```

A quantity sample type that measures the number of strokes performed while swimming.

```
static let distanceSwimming: HKQuantityTypeIdentifier
```

A quantity sample type that measures the distance the user has moved while swimming.

```
static let distanceDownhillSnowSports: HKQuantityTypeIdentifier
```

A quantity sample type that measures the distance the user has traveled while skiing or snowboarding.

```
static let basalEnergyBurned: HKQuantityTypeIdentifier
```

A quantity sample type that measures the resting energy burned by the user.

```
static let activeEnergyBurned: HKQuantityTypeIdentifier
```

A quantity sample type that measures the amount of active energy the user has burned.

```
static let flightsClimbed: HKQuantityTypeIdentifier
```

A quantity sample type that measures the number flights of stairs that the user has climbed.

```
static let nikeFuel: HKQuantityTypeIdentifier
```

A quantity sample type that measures the number of NikeFuel points the user has earned.

```
static let appleExerciseTime: HKQuantityTypeIdentifier
```

A quantity sample type that measures the amount of time the user spent exercising.

```
static let appleMoveTime: HKQuantityTypeIdentifier
```

A quantity sample type that measures the amount of time the user has spent performing activities that involve full-body movements during the specified day.

```
static let appleStandHour: HKCategoryTypeIdentifier
```

A category sample type that counts the number of hours in the day during which the user has stood and moved for at least one minute per hour.

```
static let appleStandTime: HKQuantityTypeIdentifier
```

A quantity sample type that measures the amount of time the user has spent standing.

```
static let vo2Max: HKQuantityTypeIdentifier
```

A quantity sample that measures the maximal oxygen consumption during exercise.

```
static let lowCardioFitnessEvent: HKCategoryTypeIdentifier
```

An event that indicates the user's VO2 max values consistently fall below a particular aerobic fitness threshold.

## Attachments

```
class HKAttachment
```

A file that is attached to a sample in the HealthKit store.

```
class HKAttachmentStore
```

The access point for attachments associated with samples in the HealthKit store.

```
class HKAttachmentDataReader
```

A reader that provides access to an attachment's data.

## Body measurements

```
static let height: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's height.

```
static let bodyMass: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's weight.

```
static let bodyMassIndex: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's body mass index.

```
static let leanBodyMass: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's lean body mass.

```
static let bodyFatPercentage: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's body fat percentage.

```
static let waistCircumference: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's waist circumference.

## Reproductive health

```
static let menstrualFlow: HKCategoryTypeIdentifier
```

A category sample type that records menstrual cycles.

`static let intermenstrualBleeding: HKCategoryTypeIdentifier`

A category sample type that records spotting outside the normal menstruation period.

`static let infrequentMenstrualCycles: HKCategoryTypeIdentifier`

A category sample that indicates an infrequent menstrual cycle.

`static let irregularMenstrualCycles: HKCategoryTypeIdentifier`

A category sample that indicates an irregular menstrual cycle.

`static let persistentIntermenstrualBleeding: HKCategoryTypeIdentifier`

A category sample that indicates persistent intermenstrual bleeding.

`static let prolongedMenstrualPeriods: HKCategoryTypeIdentifier`

A category sample that indicates a prolonged menstrual cycle.

`static let basalBodyTemperature: HKQuantityTypeIdentifier`

A quantity sample type that records the user's basal body temperature.

`static let cervicalMucusQuality: HKCategoryTypeIdentifier`

A category sample type that records the quality of the user's cervical mucus.

`static let ovulationTestResult: HKCategoryTypeIdentifier`

A category sample type that records the result of an ovulation home test.

`static let progesteroneTestResult: HKCategoryTypeIdentifier`

A category type that represents the results from a home progesterone test.

`static let sexualActivity: HKCategoryTypeIdentifier`

A category sample type that records sexual activity.

`static let contraceptive: HKCategoryTypeIdentifier`

A category sample type that records the use of contraceptives.

`static let pregnancy: HKCategoryTypeIdentifier`

A category type that records pregnancy.

`static let pregnancyTestResult: HKCategoryTypeIdentifier`

A category type that represents the results from a home pregnancy test.

`static let lactation: HKCategoryTypeIdentifier`

A category type that records lactation.

## Hearing

`static let environmentalAudioExposure: HKQuantityTypeIdentifier`

A quantity sample type that measures audio exposure to sounds in the environment.

`static let headphoneAudioExposure: HKQuantityTypeIdentifier`

A quantity sample type that measures audio exposure from headphones.

`static let environmentalAudioExposureEvent: HKCategoryTypeIdentifier`

A category sample type that records exposure to potentially damaging sounds from the environment.

`static let headphoneAudioExposureEvent: HKCategoryTypeIdentifier`

A category sample type that records exposure to potentially damaging sounds from headphones.

~~`static let audioExposureEvent: HKCategoryTypeIdentifier`~~

A category sample type for audio exposure events.

Deprecated

## Vital signs

`static let heartRate: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's heart rate.

`static let lowHeartRateEvent: HKCategoryTypeIdentifier`

A category sample type for low heart rate events.

`static let highHeartRateEvent: HKCategoryTypeIdentifier`

A category sample type for high heart rate events.

`static let irregularHeartRhythmEvent: HKCategoryTypeIdentifier`

A category sample type for irregular heart rhythm events.

`static let restingHeartRate: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's resting heart rate.

`static let heartRateVariabilitySDNN: HKQuantityTypeIdentifier`

A quantity sample type that measures the standard deviation of heartbeat intervals.

`static let heartRateRecoveryOneMinute: HKQuantityTypeIdentifier`

A quantity sample that records the reduction in heart rate from the peak exercise rate to the rate one minute after exercising ended.

`static let atrialFibrillationBurden: HKQuantityTypeIdentifier`

A quantity type that measures an estimate of the percentage of time a person's heart shows signs of atrial fibrillation (AFib) while wearing Apple Watch.

`static let walkingHeartRateAverage: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's heart rate while walking.

`let HKDataTypeIdentifierHeartbeatSeries: String`

A series sample containing heartbeat data.

`class HKElectrocardiogramType`

A type that identifies samples containing electrocardiogram data.

`static let oxygenSaturation: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's oxygen saturation.

`static let bodyTemperature: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's body temperature.

`static let bloodPressure: HKCorrelationTypeIdentifier`

A correlation sample that combines a systolic sample and a diastolic sample into a single blood pressure reading.

`static let bloodPressureSystolic: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's systolic blood pressure.

`static let bloodPressureDiastolic: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's diastolic blood pressure.

`static let respiratoryRate: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's respiratory rate.

## Nutrition

☰ Nutrition Type Identifiers

Type identifiers used for tracking diet and nutrition.

## Alcohol consumption

```
static let bloodAlcoholContent: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's blood alcohol content.

```
static let numberOfAlcoholicBeverages: HKQuantityTypeIdentifier
```

A quantity sample type that measures the number of standard alcoholic drinks that the user has consumed.

## Mobility

```
static let appleWalkingSteadiness: HKQuantityTypeIdentifier
```

A quantity sample type that measures the steadiness of the user's gait.

```
static let appleWalkingSteadinessEvent: HKCategoryTypeIdentifier
```

A category sample type that records an incident where the user showed a reduced score for their gait's steadiness.

```
static let sixMinuteWalkTestDistance: HKQuantityTypeIdentifier
```

A quantity sample type that stores the distance a user can walk during a six-minute walk test.

```
static let walkingSpeed: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's average speed when walking steadily over flat ground.

```
static let walkingStepLength: HKQuantityTypeIdentifier
```

A quantity sample type that measures the average length of the user's step when walking steadily over flat ground.

```
static let walkingAsymmetryPercentage: HKQuantityTypeIdentifier
```

A quantity sample type that measures the percentage of steps in which one foot moves at a different speed than the other when walking on flat ground.

```
static let walkingDoubleSupportPercentage: HKQuantityTypeIdentifier
```

A quantity sample type that measures the percentage of time when both of the user's feet touch the ground while walking steadily over flat ground.

```
static let stairAscentSpeed: HKQuantityTypeIdentifier
```

A quantity sample type measuring the user's speed while climbing a flight of stairs.

`static let stairDescentSpeed: HKQuantityTypeIdentifier`

A quantity sample type measuring the user's speed while descending a flight of stairs.

## Symptoms

☰ Symptom Type Identifiers

Identifiers for medical symptoms.

## Lab and test results

`static let bloodAlcoholContent: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's blood alcohol content.

`static let bloodGlucose: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's blood glucose level.

`static let electrodermalActivity: HKQuantityTypeIdentifier`

A quantity sample type that measures electrodermal activity.

`static let forcedExpiratoryVolume1: HKQuantityTypeIdentifier`

A quantity sample type that measures the amount of air that can be forcibly exhaled from the lungs during the first second of a forced exhalation.

`static let forcedVitalCapacity: HKQuantityTypeIdentifier`

A quantity sample type that measures the amount of air that can be forcibly exhaled from the lungs after taking the deepest breath possible.

`static let inhaledUsage: HKQuantityTypeIdentifier`

A quantity sample type that measures the number of puffs the user takes from their inhaler.

`static let insulinDelivery: HKQuantityTypeIdentifier`

A quantity sample that measures the amount of insulin delivered.

`static let numberOfTimesFallen: HKQuantityTypeIdentifier`

A quantity sample type that measures the number of times the user fell.

`static let peakExpiratoryFlowRate: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's maximum flow rate generated during a forceful exhalation.

`static let peripheralPerfusionIndex: HKQuantityTypeIdentifier`

A quantity sample type that measures the user's peripheral perfusion index.

## Mindfulness and sleep

```
static let mindfulSession: HKCategoryTypeIdentifier
```

A category sample type for recording a mindful session.

```
static let sleepAnalysis: HKCategoryTypeIdentifier
```

A category sample type for sleep analysis information.

```
HKCategoryValueSleepAnalysisAsleepValues
```

```
static let appleSleepingWristTemperature: HKQuantityTypeIdentifier
```

A quantity sample type that records the wrist temperature during sleep.

```
HKAppleSleepingBreathingDisturbancesClassificationForQuantity
```

```
enum HKAppleSleepingBreathingDisturbancesClassification
```

## Self care

```
static let toothbrushingEvent: HKCategoryTypeIdentifier
```

A category sample type for toothbrushing events.

```
static let handwashingEvent: HKCategoryTypeIdentifier
```

A category sample type for handwashing events.

## Workouts

```
let HKWorkoutTypeIdentifier: String
```

The workout type identifier.

```
let HKWorkoutRouteTypeIdentifier: String
```

A series sample containing location data that defines the route the user took during a workout.

## Clinical records

```
struct HKClinicalTypeIdentifier
```

A type identifier for the different categories of clinical records.

## UV exposure

```
static let uvExposure: HKQuantityTypeIdentifier
```

A quantity sample type that measures the user's exposure to UV radiation.

## Vision

```
let HKVisionPrescriptionTypeIdentifier: String
```

A type identifier for vision prescription samples.

## Diving

```
static let underwaterDepth: HKQuantityTypeIdentifier
```

A quantity sample that records a person's depth underwater.

```
static let waterTemperature: HKQuantityTypeIdentifier
```

A quantity sample that records the water temperature.

## Utilities

```
struct BufferedAsyncByteIterator
```

An asynchronous iterator for byte data.

---

## See Also

### 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.



## 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.