

[HealthKit](#) / [HKWorkoutEvent](#)

Class

HKWorkoutEvent

An object representing an important event during a workout.

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 13.0+ | macOS 13.0+ | visionOS 1.0+ | watchOS 2.0+

```
class HKWorkoutEvent
```

Overview

You can use workout events to toggle a workout between an active and an inactive state, or to mark points of interest during a workout.

Workouts start in an active state. A pause event switches it to an inactive state; a resume event switches it back to an active state. Adding a pause event when the workout is already inactive, or a resume event when the workout is already active, does not affect the workout's state. These events are ignored.

The lap, segment, and marker events are used to identify periods of interest during a workout. Use lap events to partition a workout into segments of equal distance. Segment events mark important periods during the workout, while markers identify important points in time.

Topics

Creating workout events

```
convenience init(type: HKWorkoutEventType, dateInterval: DateInterval,  
metadata: [String : Any]?)
```

Instantiates and returns a new workout event with the specified type, date interval, and metadata.

Getting property data

```
var dateInterval: DateInterval
```

The time and duration of the event.

```
var type: HKWorkoutEventType
```

The type of workout event.

```
var metadata: [String : Any]?
```

The metadata associated with the workout event.

Determining the event type

```
enum HKWorkoutEventType
```

Constants that represent events occurring during a workout.

Deprecated

```
convenience init(type: HKWorkoutEventType, date: Date)
```

Instantiates and returns a new workout event with the specified type and date.

Deprecated

```
convenience init(type: HKWorkoutEventType, date: Date, metadata: [String : Any])
```

Instantiates and returns a new workout event with the specified type, date, and metadata.

Deprecated

```
var date: Date
```

The time when the transition occurred.

Deprecated

Relationships

Inherits From

NSObject

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSCoding
NSCopying
NSObjectProtocol
NSSecureCoding
Sendable
SendableMetatype

See Also

Samples

- 📄 Adding samples to a workout
Create associated samples that add details to a workout.
- 📄 Accessing condensed workout samples
Read series data from condensed workouts.
- 📄 Dividing a HealthKit workout into activities
Partition multisport and interval workouts into activities that represent the different parts of the workout.

class HKWorkout

A workout sample that stores information about a single physical activity.

class HKWorkoutActivity

An object that describes an activity within a longer workout.

class HKWorkoutBuilder

A builder object that incrementally constructs a workout.

```
class HKWorkoutType
```

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

```
let HKWorkoutTypeIdentifier: String
```

The workout type identifier.

```
enum HKWorkoutActivityType
```

The type of activity performed during a workout.

```
enum HKWorkoutSessionType
```

The type of session.