Protocol

Identifiable

A class of types whose instances hold the value of an entity with stable identity.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | macOS 10.15+ | tvOS 13.0+ | visionOS 1.0+ | watchOS 6.0+

```
protocol Identifiable<ID>
```

Overview

Use the `Identifiable` protocol to provide a stable notion of identity to a class or value type. For example, you could define a `User` type with an `id` property that is stable across your app and your app's database storage. You could use the `id` property to identify a particular user even if other data fields change, such as the user's name.

`Identifiable` leaves the duration and scope of the identity unspecified. Identities can have any of the following characteristics:

- Guaranteed always unique, like UUIDs.
- Persistently unique per environment, like database record keys.
- Unique for the lifetime of a process, like global incrementing integers.
- Unique for the lifetime of an object, like object identifiers.
- Unique within the current collection, like collection indices.

It's up to both the conformer and the receiver of the protocol to document the nature of the identity.

Conforming to the Identifiable Protocol

`Identifiable` provides a default implementation for class types (using `ObjectIdentifier`), which is only guaranteed to remain unique for the lifetime of an object. If an object has a stronger notion of identity, it may be appropriate to provide a custom implementation.

Topics

Specifying the Associated Type

`associatedtype ID : Hashable`

A type representing the stable identity of the entity associated with an instance.

Required

Specifying the Identified Item

`var id: Self.ID`

The stable identity of the entity associated with this instance.

Required Default implementation provided.

Relationships

Inherited By

`DistributedActor`

Conforming Types

`Never`

See Also

Equality and Ordering

`protocol Equatable`

A type that can be compared for value equality.

```
protocol Comparable
```

A type that can be compared using the relational operators <, <=, >=, and >.