All Technologies

App Intents

AppEntity

Specifying properties

AppEntity.Property

Making the entity queryable

static var defaultQuery: Self.DefaultQ...

DefaultQuery

static var defaultResolverSpecificatio...

static var defaultResolverSpecificatio...

Default Implementations

Identifiable Implementations

FileEntity

IndexedEntity

TransientAppEntity

UniqueAppEntity

URLRepresentableEntity

Entity identity

PersistentlyIdentifiable

EntityIdentifier

EntityIdentifierConvertible

Filter /

App Intents / AppEntity

Protocol

AppEntity

An interface for exposing a custom type or app-specific concept to system experiences like Siri and the Shortcuts app.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst | macOS 13.0+ | tvOS 16.0+ | visionOS | watchOS 9.0+

```
protocol AppEntity : AppValue, DisplayRepresentable, Identifiable where Self == Self.Valu
```

Mentioned in

- Integrating actions with Siri and Apple Intelligence
- Integrating custom data types into your intents
- Adding parameters to an app intent
- Responding to the Action button on Apple Watch Ultra
- Making app entities available in Spotlight

Overview

To use a data model object to app intents, update it to conform to the `AppEntity` protocol. Declare properties using the `@Property` property wrapper to make them visible to the system. The following example from the Accelerating app interactions with App Intents sample app shows a data model for a trail:

```
struct TrailEntity: AppEntity {
    // Provide the system with the interface required to query `TrailEntity` structures.
    static let defaultQuery = TrailEntityQuery()

    // The system requires the `AppEntity` identifier to be unique and persistant because t
    var id: Trail.ID

    @Property var name: String

    @Property(title: "Region")
    var regionDescription: String

    @Property var trailLength: Measurement<UnitLength>

    var imageName: String

    var currentConditions: String

    /**
    Information on how to display the entity to people — for example, a string like the tra
    and image for a visually rich display.
    */
    var displayRepresentation: DisplayRepresentation {
        DisplayRepresentation(title: "\(name)",
                              subtitle: "\(regionDescription)",
                              image: DisplayRepresentation.Image(named: imageName))
    }

    init(trail: Trail) {
        self.id = trail.id
        self.imageName = trail.featuredImage
        self.currentConditions = trail.currentConditions
        self.name = trail.name
        self.regionDescription = trail.regionDescription
        self.trailLength = trail.trailLength
    }
}

extension TrailEntity: URLRepresentableEntity {
    static var urlRepresentation: URLRepresentation {
        // Use string interpolation to fill values from your entity necessary for construct
        // This example URL uses the unique and persistant identifier for the `TrailEntity`
        "https://example.com/trail/\(.id)/details"
    }
}
```

For additional property types, see `EntityProperty`.

It is up to you whether you want to conform to the `AppEntity` protocol directly on the data models of your app, or if you create data models specific to your app intents implementation. In many cases, it's a good idea to create models specific to app intents that shadow your app data models to keep entities separate from the rest of your app's logic.

Topics

Specifying properties

`typealias Property`

Making the entity queryable

`static var defaultQuery: Self.DefaultQuery`

The default query to use to retrieve entity property instances.

Required Default implementations provided.

`associatedtype DefaultQuery : EntityQuery`

Required

`static var defaultResolverSpecification: EmptyResolverSpecification<Self>`

`static var defaultResolverSpecification: some ResolverSpecification`

Default Implementations

Identifiable Implementations

Relationships

Inherits From

`AppValue`
`CustomLocalizedStringResourceConvertible`
`DisplayRepresentable`
`Identifiable`
`InstanceDisplayRepresentable`
`PersistentlyIdentifiable`
`Sendable`
`SendableMetatype`
`TypeDisplayRepresentable`

Inherited By

`AssistantEntity`
`AssistantSchemaEntity`
`FileEntity`
`IndexedEntity`
`TransientAppEntity`
`URLRepresentableEntity`
`UniqueAppEntity`

See Also

Entities

Integrating custom data types into your intents

Provide the system with information about the types your app uses to model its data so that your intents can use those types as parameters.

`protocol FileEntity`

An entity that refers to a document or other file.

`protocol IndexedEntity`

`IndexedEntity` represents an App Entity decorated with an attribute set. A set of attributes that enable the system to perform structured indexing and queries of entities.

`protocol TransientAppEntity`

A type that represents a transient model object which exposes its interface to App Intents via properties. Note that `TransientAppEntity` types are not meant to be queried.

`protocol UniqueAppEntity`

An entity that will only ever have one value, such as global settings.

`protocol URLRepresentableEntity`

An app entity with a URL representation.

Copyright © 2025 Apple Inc. All rights reserved. Terms of Use | Privacy Policy | Agreements and Guidelines