Documentation Language: Swift

App Intents / AppIntent

Protocol

AppIntent

An interface for providing an app-specific capability that people invoke from 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 AppIntent : PersistentlyIdentifiable, _SupportsAppDependencies, Sendable
```

Mentioned in

- Making actions and content discoverable and widely available
- Responding to the Action button on Apple Watch Ultra
- Creating your first app intent
- Integrating actions with Siri and Apple Intelligence
- Adding parameters to an app intent

Overview

To expose your app’s functionality to system experiences like Siri or the Shortcuts app, and to support interactivity in widgets, you need to implement the `AppIntent` protocol. Use it to provide phrases that can launch the functionality, describe the needed data for the functionality you make available, and implement the method that performs the functionality.

The system instantiates an app intent you create parameter-less using the `init()` initializer whenever a person invokes it through a system service like Siri, Shortcuts, and so on. If available, the system sets parameters based on user input or other available sources. With set parameters, the system attempts to resolve them in the order of their declaration in the `AppIntent` body.

After it resolves all parameters, the system calls perform() to perform the app intent with its configured parameters. Note that the system retains the app intent and its output only for the duration of the invocation.

> **Related sessions from WWDC22**
>
> Session 10032: Dive into App Intents.

Implement the AppIntent Protocol

Declare a custom intent type by defining a structure that conforms to the AppIntent protocol:

```
struct OrderSoupIntent: AppIntent {
   static var title = LocalizedStringResource("Order Soup")
   static var description = IntentDescription("Orders a soup from your favorite rest
}
```

Then, declare the AppIntent's parameters. When you implement an AppIntent type, parameters must be declared with the @Parameter property wrapper. For more information about declaring parameters, see Adding parameters to an app intent.

```
struct OrderSoupIntent: AppIntent {
   @Parameter(title: "Soup")
   var soup: Soup

   @Parameter(title: "Quantity")
   var quantity: Int?
}
```

Next, implement the required perform()function: Validate your intent's parameters, execute the intent, and return an IntentResult that represents the output of a completed intent; for example, a PerformResult.

```
struct OrderSoupIntent: AppIntent {
    @Parameter(title: "Soup")
    var soup: Soup

    @Parameter(title: "Quantity")
    var quantity: Int?
    static var parameterSummary: some ParameterSummary {
        Summary("Order \(\.$soup)") {
            \.$quantity
        }
    }

    func perform() async throws -> some IntentResult {
        guard let quantity = quantity, quantity < 10 else {
            throw $quantity.needsValue
        }
        soup.order(quantity: quantity)
        return .result()
    }
}
```

Topics

Creating an app intent

`init()`

Creates an app intent.

Required

Specifying the authentication policy

`static var authenticationPolicy: IntentAuthenticationPolicy`

A property that defines the authentication policy that indicates whether this app intent requires the device to be unlocked or otherwise authenticated.

Required Default implementation provided.

`enum IntentAuthenticationPolicy`

An enumeration that describes the authentication policy to use when running an app intent.

Configuring the metadata

`static var title: LocalizedStringResource`

A short, localized, human-readable string that describes the app intent using a verb and a noun in title case.

Required Default implementation provided.

`static var description: IntentDescription?`

A description of the app intent that the system shows to people.

Required Default implementation provided.

~~`static var openAppWhenRun: Bool`~~

A boolean property that tells the system to consider the app intent even if its app is not in the foreground.

Required Default implementations provided.

Deprecated

`static var isDiscoverable: Bool`

A boolean value that determines whether system features such as Shortcuts and Spotlight can discover this app intent.

Required Default implementation provided.

Performing the action

`func perform() async throws -> Self.PerformResult`

Performs the intent after resolving the provided parameters.

Required Default implementations provided.

`var systemContext: IntentSystemContext`

Context information that’s available while the system performs the app intent’s action.

`associatedtype PerformResult : IntentResult`

Required

Requesting confirmation

`func requestConfirmation() async throws`

Requests user confirmation before performing the app intent.

`func requestConfirmation(conditions: ConfirmationConditions, actionName: ConfirmationActionName, dialog: IntentDialog) async throws`

Requests user confirmation before performing the app intent.

`func requestConfirmation<Content>(conditions: ConfirmationConditions, actionName: ConfirmationActionName, dialog: IntentDialog?, showDialogAsPrompt: Bool, content: () -> Content) async throws`

Request user confirmation before performing the app intent.

~~func requestConfirmation<Result>(result: Result, confirmationActionName: ConfirmationActionName, showPrompt: Bool) async throws~~

Requests user confirmation before performing the app intent.

Deprecated

~~func requestConfirmation<Result>(output: Result, confirmationActionName: ConfirmationActionName, showPrompt: Bool) async throws~~

Deprecated

Donating the intent to the system

```
func donate() async throws -> IntentDonationIdentifier
```

Donates the intent to the transcript.

```
func donate() -> IntentDonationIdentifier
```

Donates the intent to the transcript.

```
func donate(result: some IntentResult) async throws -> IntentDonationIdentifier
```

Donates the intent and optional result to the transcript.

```
func donate(result: some IntentResult) -> IntentDonationIdentifier
```

Donates the intent and optional result to the transcript.

```
func callAsFunction(donate: Bool) async throws -> Self.PerformResult.Value
```

```
func callAsFunction(donate: Bool) async throws
```

Summarizing the parameters

```
associatedtype SummaryContent : ParameterSummary
```

The type of parameter summary representing this intent.

Required

```
static var parameterSummary: Self.SummaryContent
```

Defines the summary of this intent in relation to how its parameters are populated.

Required Default implementation provided.

```
static var parameterSummary: some ParameterSummary
```

```
enum ParameterSummaryBuilder
```

A result builder that allows you to declaratively describe a parameter summary.

```
typealias Parameter
```

```
typealias Case
```

```
typealias DefaultCase
```

```
typealias Summary
```

```
typealias Switch
```

```
typealias When
```

Instance Methods

```
func continueInForeground(IntentDialog?, alwaysConfirm: Bool) async throws
```

A method you call to ask a person to continue an action in the foreground.

```
func needsToContinueInForegroundError(IntentDialog?, alwaysConfirm: Bool) -> AppIntentError
```

A method you call to ask a person to continue an intent’s action in the foreground after it encounters an error.

```
func requestChoice(between: [IntentChoiceOption], dialog: IntentDialog?) async throws -> IntentChoiceOption
```

Pauses the app intent to request a person to choose from several options.

```
func requestChoice<Content>(between: [IntentChoiceOption], dialog: IntentDialog?, content: () -> Content) async throws -> IntentChoiceOption
```

Pauses the app intent to request a person to choose from several options.

```
func requestChoice<Content>(between: [IntentChoiceOption], dialog: IntentDialog?, view: Content) async throws -> IntentChoiceOption
```

Pauses the app intent to request a person to choose from several options.

```
func requestConfirmation<Snippet>(conditions: ConfirmationConditions, actionName: ConfirmationActionName, dialog: IntentDialog?, showDialogAsPrompt: Bool, snippetIntent: Snippet) async throws
```

```
func requestConfirmation<Snippet>(conditions: ConfirmationConditions, actionName: ConfirmationActionName, dialog: IntentDialog?, showDialogAsPrompt: Bool, snippetIntent: Snippet) async throws -> Snippet.PerformResult.Value
```

Requests user confirmation before performing the app intent.

Type Aliases

`typealias Option`

A convenience type alias that represents a choice option within the scope of an app intent.

Type Properties

`static var supportedModes: IntentModes`

Defines the supported modes that describe the behavior of your app intent.

Required Default implementation provided.

Relationships

Inherits From

`PersistentlyIdentifiable`, `Sendable`, `SendableMetatype`

Inherited By

`AssistantIntent`
`AssistantSchemaIntent`
`AudioPlaybackIntent`
`AudioRecordingIntent`
`AudioStartingIntent`
`CameraCaptureIntent`
`ControlConfigurationIntent`
`CustomIntentMigratedAppIntent`
`DeleteIntent`
`DeprecatedAppIntent`
`ForegroundContinuableIntent`
`LiveActivityIntent`
`LiveActivityStartingIntent`
`OpenIntent`
`PauseWorkoutIntent`
`PlayVideoIntent`
`PredictableIntent`
`ProgressReportingIntent`

`PushToTalkTransmissionIntent`
`ResumeWorkoutIntent`
`SetFocusFilterIntent`
`SetValueIntent`
`ShowInAppSearchResultsIntent`
`SnippetIntent`
`StartDiveIntent`
`StartWorkoutIntent`
`SystemIntent`
`TargetContentProvidingIntent`
`UISceneAppIntent`
`URLRepresentableIntent`
`UndoableIntent`
`WidgetConfigurationIntent`

Conforming Types

`EmptySnippetIntent`, `OpenURLIntent`

See Also

Actions

`protocol AudioPlaybackIntent`

An App Intent that plays, pauses, or otherwise modifies audio playback state when it executes.

`protocol AudioRecordingIntent`

An app intent that starts, stops or otherwise modifies audio recording state.

~~`protocol AudioStartingIntent`~~

An App Intent that plays, pauses, or otherwise modifies audio playback state when it executes.

Deprecated

`protocol CameraCaptureIntent`

Designates intent that will launch an activity that uses device’s camera to capture photos or videos. Marking your intent with this protocol makes it available as a possible action for Camera quick action.

`protocol DeleteIntent`

Delete the associated entity(s).

`protocol DeprecatedAppIntent`

An app intent that marks an action as deprecated and informs people which action to use instead.

~~`protocol ForegroundContinuableIntent`~~

A protocol you use for app intents which begin their work with the app in the background but may request to continue in the foreground.

Deprecated

`protocol OpenIntent`

Open the associated item.

`struct OpenURLIntent`

An intent that opens a universal link.

`protocol PlayVideoIntent`

An intent that looks for videos based on a search term, then plays the content.

`protocol ProgressReportingIntent`

An intent that reports progress to the system during its execution

`protocol PushToTalkTransmissionIntent`

An intent that begins or ends an audio transmission with the Push to Talk framework.

`protocol URLRepresentableIntent`

An app intent with a URL representation.

`protocol SetValueIntent`

An intent that contains a value which can be set.

`protocol ShowInAppSearchResultsIntent`

An app intent that takes a person to search results for a specified search term.