

[Intents](#) / INIntentHandlerProviding

## Protocol

# INIntentHandlerProviding

The interface for dispatching intents to the custom objects that handle those intents.

iOS 10.0+ | iPadOS 10.0+ | Mac Catalyst 13.1+ | macOS 11.0+ | tvOS 14.0+ | visionOS 1.0+ | watchOS 3.2+

```
protocol INIntentHandlerProviding : NSObjectProtocol
```

## Overview

The `INIntentHandlerProviding` protocol defines the method for routing SiriKit requests to the appropriate objects of your Intents extension. The method in this protocol acts as a dispatcher, returning an object capable of resolving, confirming, and handling a specific intent object. This protocol is adopted by the `INExtension` class and you must provide a custom implementation.

When you add an Intents extension to your project, Xcode automatically creates a default `INExtension` subclass that adopts this protocol. Modify the `handler(for:)` method in that subclass and use it to create your handler objects. You can use the same class to handle multiple intents or you can use different classes for each intent.

## Topics

### Providing Handlers for Intents

```
func handler(for: INIntent) -> Any?
```

Returns the object capable of handling the specified intent in an extension.

# Relationships

## Inherits From

`NSObjectProtocol`

## Conforming Types

`INExtension`


---

# See Also

## Intents App Extension

 [Creating an Intents App Extension](#)

Add and configure an Intents app extension in your Xcode project.

 [Structuring Your Code to Support App Extensions](#)

Move your back-end services to a private framework so your app and app extensions can use them.

`class` `INExtension`

The entry point for an Intents extension.