Instance Method

handler(for:)

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

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

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

Required

Parameters

`intent`

The intent object representing the request coming from the system.

Return Value

The object capable of handling the specified intent, or `nil` if your app doesn't handle the intent. Return an object that conforms to the protocol for handling intents of the same type as the provided `intent`.

Mentioned in

Dispatching intents to handlers

Discussion

This method acts as a dispatcher, mapping incoming intents to the object capable of handling them. In your implementation, check the `intent` parameter's type and return an object capable of handling that type. As SiriKit invokes this method on an arbitrary queue, limit your implementation to just resolving the handler, and dispatch additional work to an alternate queue.

The object you return must adopt the handling protocol for the specified intent. For example, when receiving an `INSendMessageIntent` object, return an object that adopts `INSendMessageIntentHandling`. The simplest way to implement this method is to check the `intent` parameter's type and return the appropriate object, as shown in the following example.

Swift Objective-C

```
override func handler(for intent: INIntent) -> Any? {
    switch intent {
    case is INSendMessageIntent:
        return SendMessageHandler()
    case is INSearchForMessagesIntent:
        return SearchForMessagesHandler()
    default:
        return nil
    }
}
```

You're responsible for providing the handler objects and for ensuring they adopt the necessary protocols to respond to a given intent. In the example above, the `SendMessageHandler` class implements the `INSendMessageIntentHandling` protocol, and the `SearchForMessagesHandler` class implements the `INSearchForMessagesIntentHandling` protocol.

When creating handler objects in this method, use the provided `intent` object only to determine which handler to create. Don't use the `intent` object to initialize your handler object and don't store a reference to the intent object for later use. SiriKit updates the intent object during the processing of the request to incorporate any new information provided by the user.

The system calls this `INExtension` method for intents listed in the Supported Intents section of an extension's target. To handle intents in your app directly, see `application(_:handlerFor:)`.

For information about how to handle intents, see Resolving and Handling Intents.