

□ Documentation

[SiriKit / Confirming the Details of an Intent](#)

Article

Confirming the Details of an Intent

Perform final validation of the intent parameters and verify that your services are ready to fulfill the intent.

Overview

After you resolve the parameters of an intent, but before you handle that intent, SiriKit may ask you to confirm the intent details. The confirmation phase is your chance to perform any final validation of the intent parameters and to verify that any needed services are available. For example, you might confirm that you can communicate with your company's server. After performing your final validation, you provide SiriKit with a response object indicating how you plan to handle the request.

The listing below shows a confirmation method used by a running app to validate an [INStartWorkoutIntent](#) object. Because the app uses Core Location to map the user's run, the method verifies that the needed location services are available and returns a failure code if they are not. Otherwise, the response object contains the [INStartWorkoutIntentResponseCode.ready](#) code, which indicates that the app is ready to start the run.

```
func confirm(startWorkout intent: INStartWorkoutIntent,  
            completion: @escaping (INStartWorkoutIntentResponse) -> Void) {  
    var code : INStartWorkoutIntentResponseCode = .ready  
    var activity : NSUserActivity? = nil  
  
    // If location services are not available, the user  
    // might need to change the type of workout.  
    if !CLLocationManager.locationServicesEnabled() {  
        code = .failureRequiringAppLaunch  
        activity = NSUserActivity(activityType:  
            "com.example.myRunningApp.noLocationAvailable")  
    }  
}
```

```
// Deliver the response back to SiriKit.  
let response = INStartWorkoutIntentResponse(code: code,  
    userActivity: activity)  
completion(response)  
}
```

Providing a custom `NSUserActivity` object with your response during confirmation is recommended but not required. If SiriKit needs to launch your app for any reason, it passes the provided activity object to your app. Your app can use any information in the object to take further actions. For example, you might include custom data in the user activity object that your app can then use to configure its interface.

See Also

Articles

-  [Adding User Interactivity with Siri Shortcuts and the Shortcuts App](#)
Add custom intents and parameters to help users interact more quickly and effectively with Siri and the Shortcuts app.
-  [Defining Relevant Shortcuts for the Siri Watch Face](#)
Inform Siri when your app's shortcuts may be useful to the user.
-  [Deleting Donated Shortcuts](#)
Remove your donations from Siri.
-  [Dispatching intents to handlers](#)
Provide SiriKit with an intent handler capable of handling a specific intent.
-  [Improving Siri Media Interactions and App Selection](#)
Fine-tune voice controls and improve Siri Suggestions by sharing app capabilities, customized names, and listening habits with the system.
-  [Improving interactions between Siri and your messaging app](#)
Donate app-specific content, use Siri's contact suggestions, and adopt the latest platform features to create a more consistent messaging experience.
-  [Registering Custom Vocabulary with SiriKit](#)
Register your app's custom terminology, and provide sample phrases for how to use your app with Siri.

-  **Handling an Intent**
Fulfill the intent and provide feedback to SiriKit about what you did.
-  **Resolving the Parameters of an Intent**
Validate the parameters of an intent and make sure that you have the information you need to continue.
-  **Generating a List of Ride Options**
Generate ride options for Maps to display to the user.
-  **Handling the Ride-Booking Intents**
Support the different intent-handling sequences for booking rides with Shortcuts or Maps.
-  **Donating Reservations**
Inform Siri of reservations made from your app.
-  **Specifying Synonyms for Your App Name**
Provide alternative names for your app that are more familiar or easier for users to speak.
-  **Intent Phrases**
The keys that you include in your global vocabulary file to show how users engage your app from Siri.
-  **Localizing Your Vocabulary for Chinese Dialects**
Apply emphasis markers to your pronunciation tips to assist Siri with Chinese dialects.