Article

Defining Relevant Shortcuts for the Siri Watch Face Deprecated

Inform Siri when your app’s shortcuts may be useful to the user.

watchOS 5.0–11.0 Deprecated

Overview

The Siri watch face can suggest shortcuts to the user based on a situation, time of day, or location. You determine which actions in your app are pertinent to the user and may be something they’d like to do in the future, such as starting a workout when arriving at a gym. Your app tells Siri about these actions by specifying them as relevant shortcuts.

An iOS app can set relevant shortcuts and display them on the Siri watch face, whether it has a companion watchOS app or not. If a watchOS app and its iOS companion both set relevant shortcuts, the Siri watch face uses the list of shortcuts set most recently across either app.

Important

The Siri Watch Face is available in watchOS 11 and earlier. To make content available in the Smart Stack on Apple Watch using widgets, refer to WidgetKit, Increasing the visibility of widgets in Smart Stacks, and App Intents. For more information about migrating your SiriKit code to App Intents, refer to Migrating widgets from SiriKit Intents to App Intents and Soup Chef with App Intents: Migrating custom intents.

Create a Relevant Shortcut

To let Siri know about relevant shortcuts for your app, start by creating an `INIntent` with a system-provided or custom intent, or an `NSUserActivity` object for each action. Next, create an `INShortcutReference` object for each intent and user activity. Use the shortcut to create an

INRelevantShortcutRole object, and set the shortcutRole to provide a hint to Siri as to the purpose of the shortcut: to perform an action or display information.

```
// Add an intent to the list of suggestions. To create
// a shortcut from an intent, the intent must be valid.
if let shortcut = INShortcut(intent: startWorkoutIntent) {

    let relevantShortcut = INRelevantShortcut(shortcut: shortcut)
    relevantShortcut.shortcutRole = INRelevantShortcutRole.action

}
```

Add Relevance Providers

To give Siri a hint about when to suggest the shortcut to the user, add one or more providers to the relevanceProviders property on the relevant shortcut. Relevance providers let you specify parameters such as a particular situation, date, time, and location that indicate the shortcut may be relevant to the user. For example, you can instruct Siri to suggest starting a workout when the user arrives at the gym.

```
// Add an intent to the list of suggestions. To create
// a shortcut from an intent, the intent must be valid.
if let shortcut = INShortcut(intent: startWorkoutIntent) {

    let relevantShortcut = INRelevantShortcut(shortcut: shortcut)
    relevantShortcut.shortcutRole = INRelevantShortcutRole.action

    // Create a location provider for Apple Park Fitness Center.
    let location = CLLocationCoordinate2D(latitude: 37.336_971, longitude: -122.013_
    let region = CLCircularRegion(center: location, radius: 0.1, identifier: "Apple
    let locationProvider = INLocationRelevanceProvider(region: region)
    relevantShortcut.relevanceProviders = [locationProvider]

}
```

Don't combine an INDailyRoutineRelevanceProvider for a time-based situation with an INDateRelevanceProvider, like INDailyRoutineRelevanceProvider.Situation.evening and 8 p.m.; the outcome can be unpredictable. Instead, provide multiple relevant shortcuts for each scenario, that is, provide a relevant shortcut for INDailyRoutineRelevanceProvider.Situation.evening and a second one for 8 p.m.

Applying multiple locations relevance providers to a relevant shortcut creates similar problems. Instead, provide a relevant shortcut for each location.

Important

Your app should donate an intent or user activity as a shortcut each time the user performs an action in your app, even when your app provides the shortcut as a relevant shortcut. The donation helps the system determine when it should suggest the relevant shortcut to the user. For more information about donations, see Donating Shortcuts.

Set Relevant Shortcuts

Continue creating the relevant shortcuts for your app, storing each one in an array. After you've created and stored the shortcuts, inform Siri of the relevant shortcuts by using the `default` relevant shortcut store to call `setRelevantShortcuts(_:completionHandler:)`, passing in the array of relevant shortcuts.

```
var relevantShortcuts:[INRelevantShortcut] = []

// Add an intent to the list of suggestions. To create
// a shortcut from an intent, the intent must be valid.
if let shortcut = INShortcut(intent: startWorkoutIntent) {

    let relevantShortcut = INRelevantShortcut(shortcut: shortcut)
    relevantShortcut.shortcutRole = INRelevantShortcutRole.action

    // Create a location provider for Apple Park Fitness Center.
    let location = CLLocationCoordinate2D(latitude: 37.336_971, longitude: -122.013_
    let region = CLCircularRegion(center: location, radius: 0.1, identifier: "Apple 
    let locationProvider = INLocationRelevanceProvider(region: region)
    relevantShortcut.relevanceProviders = [locationProvider]

    relevantShortcuts.append(relevantShortcut)
}

INRelevantShortcutStore.default.setRelevantShortcuts(relevantShortcuts) { (error) ir
    if let error = error {
        print("Failed to set relevant shortcuts. \(error))")
    } else {
        print("Relevant shortcuts set.")
    }
}
```

Replace Relevant Shortcuts

There isn't a way to add relevant shortcuts to the list already set by your app. Instead, you must replace the list by following the same steps previously mentioned:

- Gather the actions (as user activities or intents) and create an array of `INRelevantShortcut` objects.
- Pass the array to `setRelevantShortcuts(_:completionHandler:)`.

When you've completed these steps, the new set of relevant shortcuts takes the place of the previous one. If you want to delete all relevant shortcuts defined by your app, call `setRelevantShortcuts(_:completionHandler:)`, passing in an empty array.

Replace the list when your app needs to add or remove a relevant shortcut. It isn't necessary to replace the list more than once a day, and resubmitting the same set of relevant shortcuts more often provides no benefits.

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.

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.

Confirming the Details of an Intent

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

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.