

API Collection

App Shortcuts

Integrate your app's intents and entities with the Shortcuts app, Siri, Spotlight, and the Action button on supported iPhone and Apple Watch models.

Overview

Starting with iOS 15 and macOS 13, you can create a preconfigured App Shortcut that allows people to discover and run your app intent without any configuration. By creating App Shortcuts, you make your app's functionality instantly available for use in Shortcuts, Spotlight, and Siri from the moment a person installs your app — without any setup in the Shortcuts app or an Add to Siri button. On iPhone models that support the Action button, people can associate your preconfigured App Shortcut on the Action button for quick access of your app's functionality.

Note

Apple may extract anonymized App Shortcuts data such as localized phrases, display representation values, and the title and description of related intents. Machine learning models use this data when training to help improve the App Shortcuts experience.

Key app functionalities that people use to complete a task quickly and that you expose to the system with app intents are great candidates for App Shortcuts. For each high-value app intent, create an App Shortcut that specifies the intended action, the required parameters, the spoken phrases someone uses to run it, and the short title and the image that appear in the Shortcuts app.

To offer an App Shortcut:

1. Create an app intent for a key app functionality as described in [Providing your app's capabilities to system services](#).
2. Create the `AppShortcut` object for your app intent using the `init(intent:phrases:shortTitle:systemImageName:)` initializer with phrases people can use to run the app intent and with the metadata that appears in the Shortcuts app.
3. Implement the `AppShortcutsProvider` protocol that provides the App Shortcuts you offer to the Shortcuts app.

With these three steps, you make your app's functionality more discoverable and allow people to interact with your app in a lightweight way. However, the system displays a default interface for your App Shortcut. To display a custom view for each shortcut, return a SwiftUI view in your app intent's `perform()` method.

Related sessions from WWDC22

Session 10170: Implement App Shortcuts with App Intents and Session 10169: Design App Shortcuts.

Offer App Shortcuts with preconfigured parameters

With App Shortcuts, you can also preconfigure phrases for app intents that use specific parameters. When you include parameters, people can use one phrase to start an interaction with an app without Siri having to ask for clarification. For example, a meditation app could offer an app intent to start a meditation with the phrase "Start a meditation". Because the app offers many different meditations, Siri would require an additional clarification which meditation a person wants to start.

With an App Shortcut, you can supply preconfigured parameters ahead of time that allow a person to skip this clarification step. For example, the meditation app could provide parameterized phrases where each phrase represents a common meditation. A person could then start a meditation with one phrase like "Start a mindfulness meditation." or "Start a short meditation."

Make your App Shortcuts even more discoverable

Although App Shortcuts don't require a person to do any configuration in the Shortcuts app or by using the Add to Siri button, you may want to present elements in your app to tell people about an available App Shortcut. You have two options:

- `SiriTipView` and `SiriTipUIWebView` present a view that tells a person that an App Shortcut is available.
- `ShortcutsLink` allows you to display a link to your App Shortcut.

`ShortcutsLink` is especially convenient if your app displays a list of its available App Shortcuts.

Topics

App Shortcut management

`protocol AppShortcutsProvider`

A type alias for the type that provides an app's preconfigured shortcuts.

App Shortcut definition

`struct AppShortcut`

A type that defines a preconfigured shortcut for a specific app intent.

`struct AppShortcutPhrase`

A spoken phrase that causes the system to run the corresponding App Shortcut.

`struct NegativeAppShortcutPhrase`

An object that represents a negative phrase.

`struct NegativeAppShortcutPhrases`

This is a set of negative phrases, which will all be added to the app-level negative training set. All the training data specified here, will be used to completely bypass your app

property list key `NSAppIconActionTintColorName`

The tint color to apply to text and symbols in the App Shortcuts platter.

property list key `NSAppIconComplementingColorNames`

The names of the colors to use for the background of the App Shortcuts platter.

`enum AppShortcutsBuilder`

A result builder that allows you to declaratively describe the App Shortcuts that your app provides.

`enum ShortcutTileColor`

Describes the colors a shortcut tile in the Shortcuts app.

App Shortcut options

`struct AppShortcutOptionsCollection`

Represents a collection of options available for an `AppShortcutParameterPresentation`.

`protocol AppShortcutOptionsCollectionProtocol`

`protocol AppShortcutOptionsCollectionSpecification`

`enum AppShortcutOptionsCollectionSpecificationBuilder`

App Shortcut parameter presentation

`struct AppShortcutParameterPresentation`

Describes the presentation of an `AppShortcut` for the provided parameter.

`struct AppShortcutParameterPresentationSummary`

This struct represents the summary of an `AppShortcutParameterPresentation`, which should have the summary string. The summary should include the parameter in the interpolation, e.g. "Call (.\$person)".

`struct AppShortcutParameterPresentationSummaryString`

~~`struct AppShortcutParameterPresentationTitle`~~

This struct represents the title of an `AppShortcutParameterPresentation`, which should have both a specific and a generic title. The specific title should include the parameter in the interpolation, e.g. "Call (.\$person)"; while the generic title is a simple string that doesn't have the parameter specified, e.g. "Call Person...".

Deprecated

~~struct AppShortcutParameterPresentationTitleString~~

Deprecated

Buttons

class ShortcutsUIButton

A button that opens the current app's page in the Shortcuts app.

struct ShortcutsLink

A button that brings users to the current app's App Shortcuts page in the Shortcuts app.

struct ShortcutsLinkStyle

The styles to apply to buttons you use to open your app's page in the Shortcuts app.

Tip views

class SiriTipUIView

A view that displays the phrase a person uses to invoke an App Shortcut.

struct SiriTipView

A SwiftUI view that displays the phrase someone uses to invoke an App Shortcut.

struct SiriTipViewStyle

The styles to apply to the tip views you use to display spoken phrases.

See Also

System integration

☰ Intent discovery

Donate your app's intents to the system to help it identify trends and predict future behaviors.

☰ Focus

Adjust your app's behavior and filter incoming notifications when the current Focus changes.

☰ Action button on iPhone and Apple Watch

Enable people to run your App Shortcuts with the Action button on iPhone or to start your app's workout or dive sessions using the Action button on Apple Watch.