

[SiriKit](#) / [IntentsUI](#) / Creating an Intents UI Extension

Creating an Intents UI Extension

Create an Intents UI app extension to customize the interfaces displayed by Siri and Maps.

Overview

For some intents, Siri and Maps display details of the response provided by your Intents app extension. When displaying this information, Siri and Maps place the data that you provided into a standard system interface. For example, the default ride-booking interface includes a map of the user's location and information about the booked ride. You can customize this interface using an Intents UI app extension, which is a peer of the Intents app extension that you use to handle intents.

SiriKit offers different levels of customization for its interface:

- In iOS 11 and later, you can use intent parameters to customize every portion of the Siri or Maps interface, making it possible to replace the default interface entirely.
- You can provide a single custom view controller for SiriKit to display with the rest of the default interface.

Note

watchOS doesn't support Intents UI app extensions.

Topics

Configuration

Configuring Your Intents UI App Extension Target

Configure your Xcode project to include an Intents UI app extension that you use to customize the Siri and Maps interfaces.

Configuring the View Controller for Your Custom Interface

Configure your view controller to replace or augment the default interface in Siri or Maps.

See Also

Custom UI for Siri and Maps

`protocol INUIHostedViewControlling`

Methods for presenting custom content in the Siri and Maps interfaces.

`protocol INUIHostedViewSiriProviding`

Methods for hiding portions of the default interfaces that Siri provides.

`class INParameter`

A parameter of an interaction object.

`protocol INIntentSetImagePath`