Article

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.

Overview

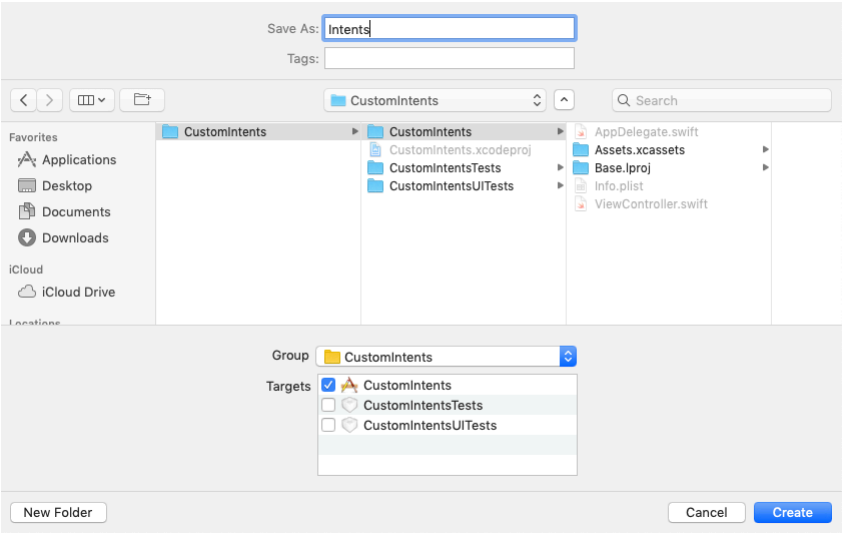
You can offer your app's unique capabilities throughout the system by designing custom intents. You may also want to create a custom intent that provides the same functionality as a system intent, to offer users more flexibility for incorporating that functionality into a multi-step shortcut.

Define custom intents and their parameters in an intents definition file. Xcode uses this file to generate an `INIntent` subclass and related data types for each of your intents.

When the user invokes your shortcut with their voice, Siri starts a dialog to collect the additional information needed to complete the shortcut. In the Shortcuts app, the user can make changes to what the shortcut will do when it's invoked. To get an app that integrates custom intents and parameters, see Soup Chef: Accelerating App Interactions with Shortcuts and download the Soup Chef sample app.

Add an Intent Definition File to Your App Target

Choose File > New > File to add a new SiriKit Intent Definition File to your Xcode project. Make sure that you select your app target.

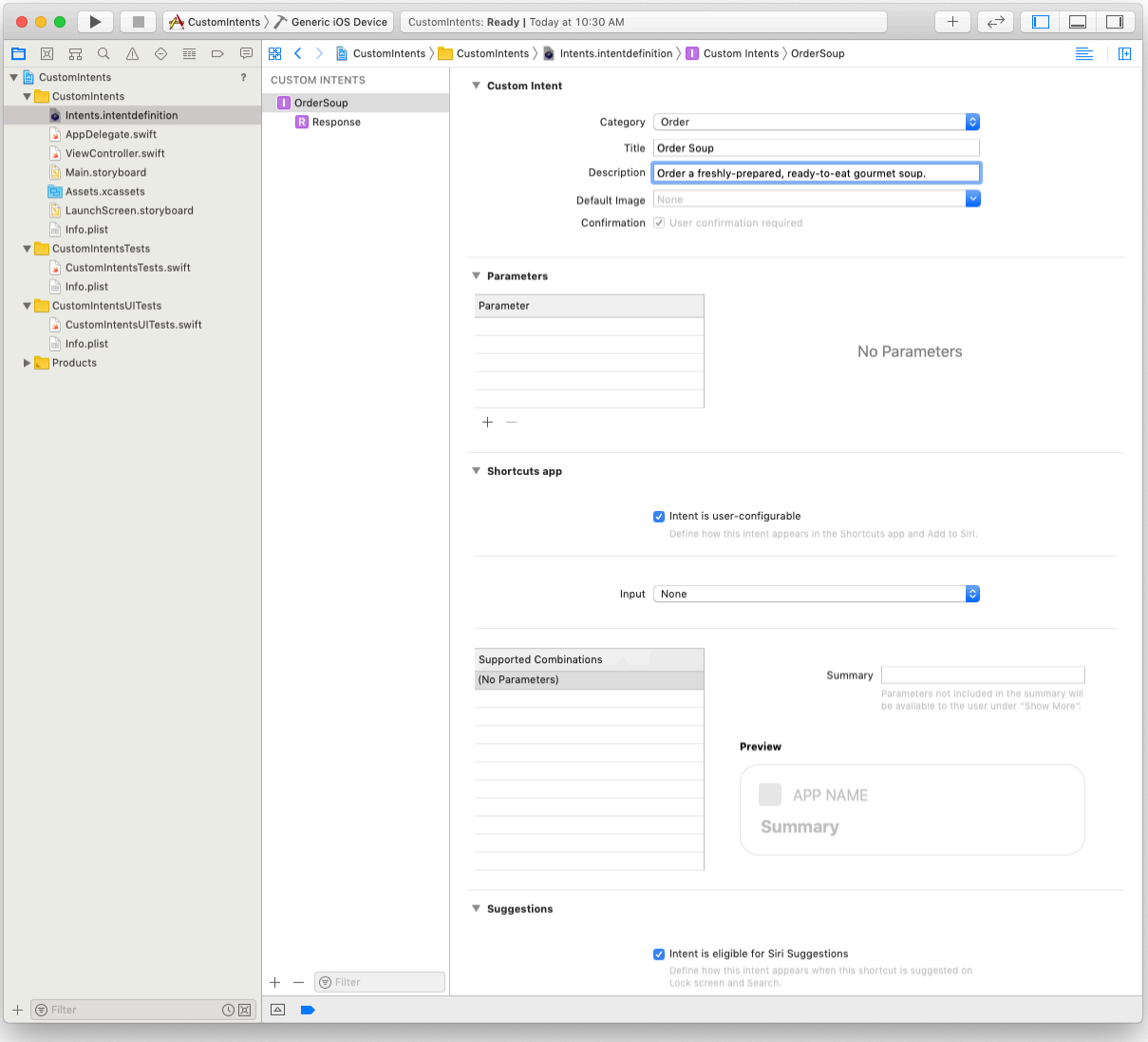


Define Your Intents

Create the new intent and define what the intent does.

1. Select the created `Intents.intentdefiniton` file.
2. In the Xcode Menu Bar, choose Editor > New Intent.
3. Name your intent using a VerbNoun convention, like SetAlarm, CheckOrderStatus, and so on.
4. Make sure the category you select matches your intent's purpose because the chosen category defines the default Siri dialog, as you will see in a later section.
5. In the title field, add a human-readable title for the intent.
6. In the Description field, add a short sentence that describes what the intent does.

The figure below shows a custom intent named OrderSoup with the category, title, and description set.

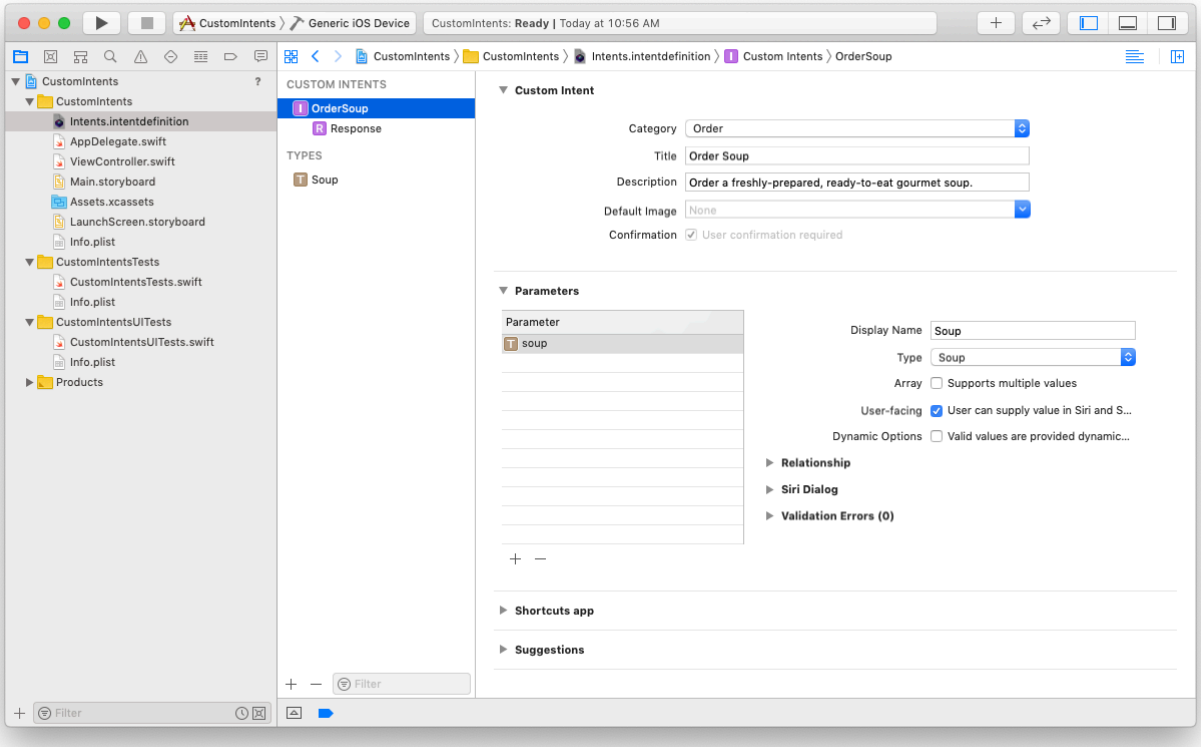


> **Note**
>
> Xcode automatically creates the following response codes for a custom intent: `unspecified`, `ready`, `continueInApp`, `inProgress`, `success`, `failure`, and `failureRequiringAppLaunch`.

Define the Intent Parameters

Before you can use the intent, you need to define the intent's parameters such as soup, quantity, and so on. Create a new parameter by clicking the plus button in the Parameters field and selecting either a system parameter type or a custom type in the Type field. System types are predefined by the system, and custom types allow you to create your own types. Parameter names must start with a lowercase character while custom type names must start with an uppercase character.

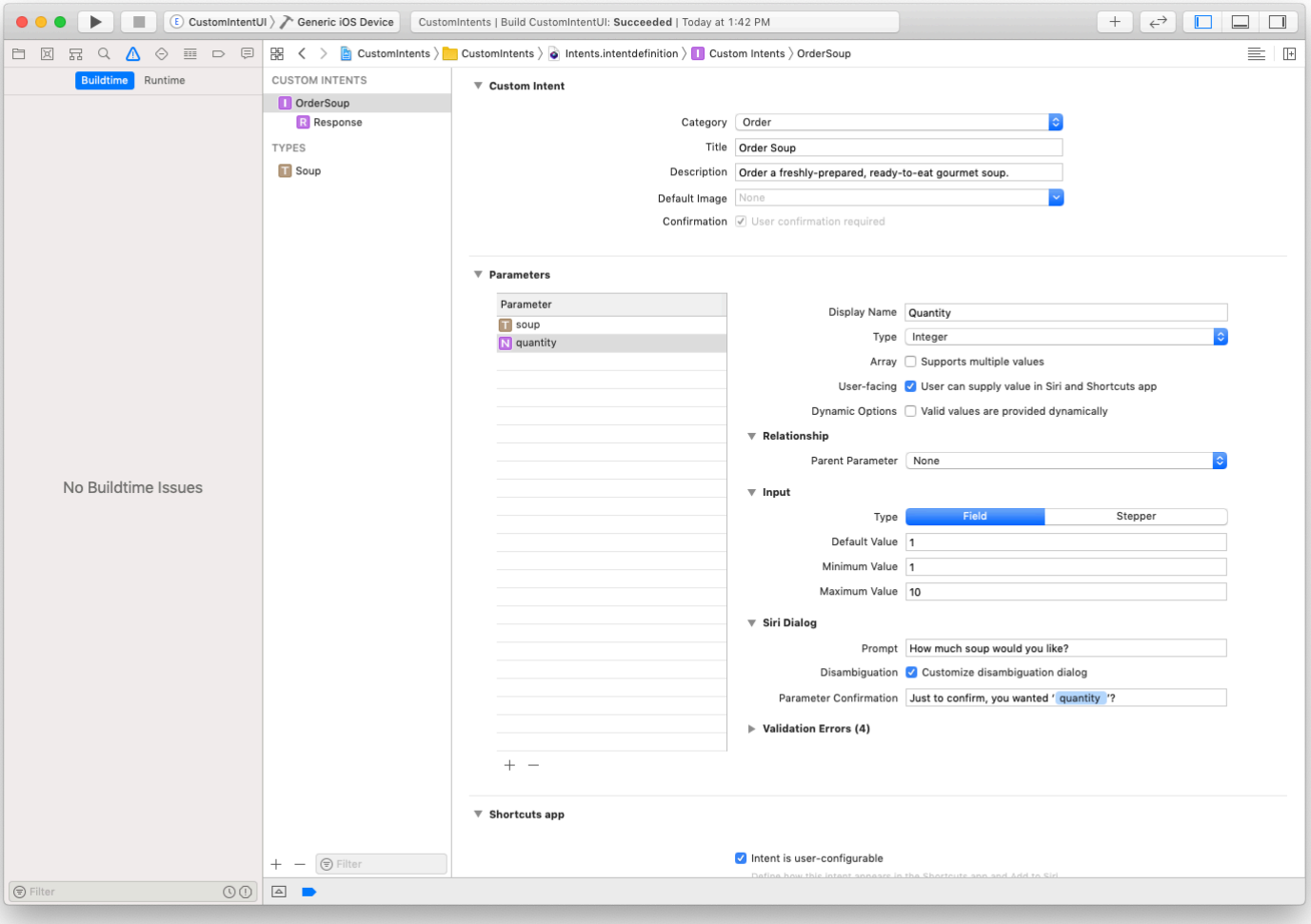
By default, Xcode selects the User-facing checkbox. This controls whether the parameter is configurable in the Shortcuts app and resolvable at runtime in both Siri and the Shortcuts app.



Add Parameter Metadata and Siri Dialog Data

Siri can prompt users to complete shortcuts that require additional input based on the parameter metadata you provide. Set the Input fields for each parameter that you want the user to have control over, such as allowing them to order different quantities of soup through a shortcut.

You can also customize the Siri dialog for each parameter resolution result. Enter the phrase you want Siri to say to the app user in the Prompt field.



Establish Relationships Between Parameters

Use relationships between parameters to determine whether certain parameters should be prompted by voice in Siri based on their association with their parent parameter. To set the parent parameter:

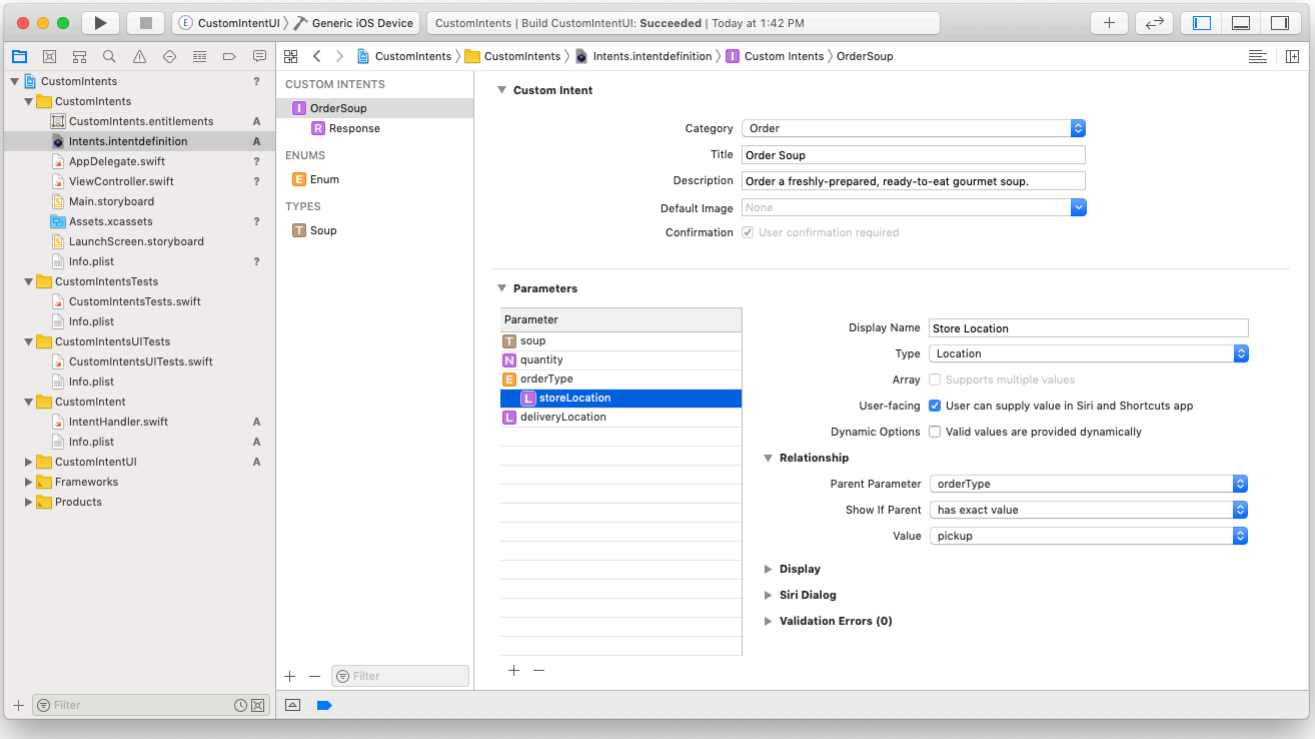
1. Expand the Relationship section.

2. Select the desired parameter from the Parent Parameter drop-down.

3. Select when to show the parameter from the Show If Parent drop-down.

4. If the parent has to have an exact value, select the value from the Value drop-down.

The following example shows the `storeLocation` parameter has a child relationship to the `orderType` parent parameter. The `orderType` parameter is an enum with two values, `pickup` and `delivery`. When the user tells Siri they will pick up their order, the `storeLocation` value is automatically used.



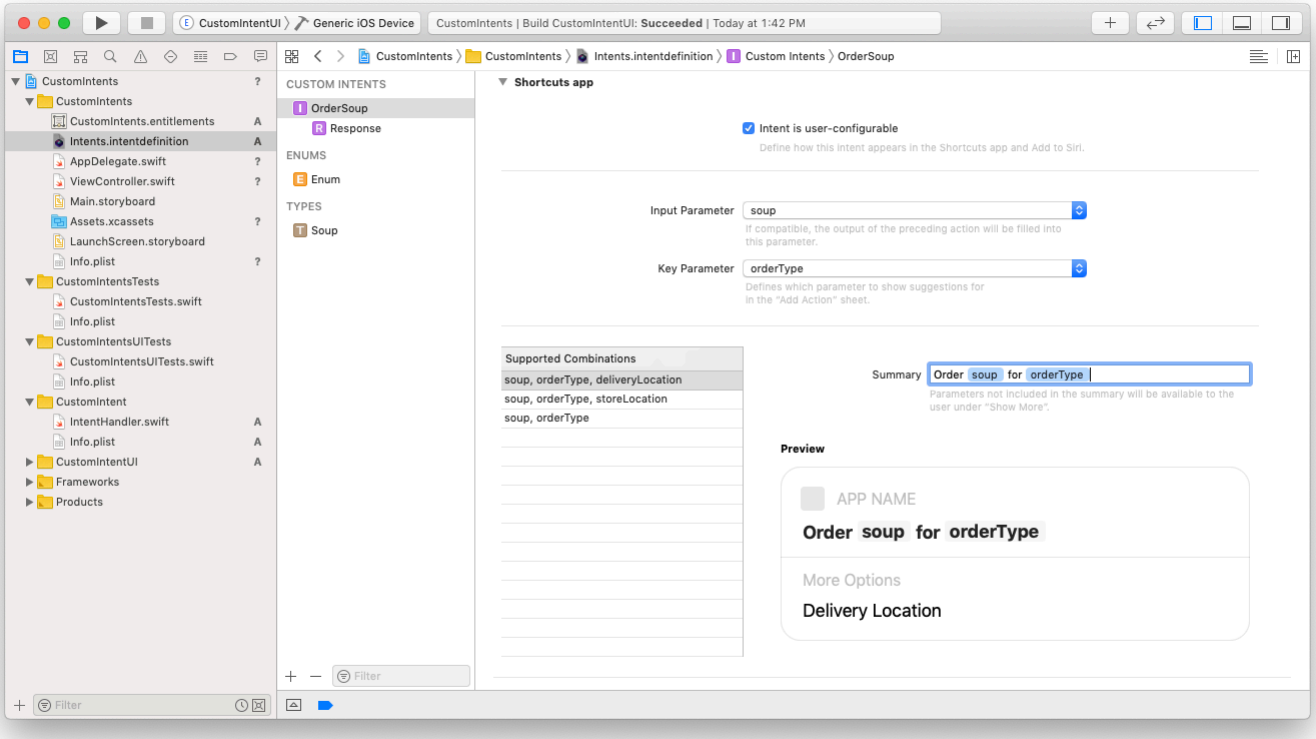
Define User-Configurable Shortcuts

User-configurable shortcuts are shortcuts that the user can configure in the Shortcuts app, and use interactively in Siri. To enable a shortcut for user configuration:

1. Select the Intent is user-configurable checkbox.

2. Fill in the Summary field.

The following example shows a Shortcut that supports the `soup`, `orderType`, and `deliveryLocation` parameters. The summary contains the `soup` and `orderType` parameters with the `deliveryLocation` in the More Options section.

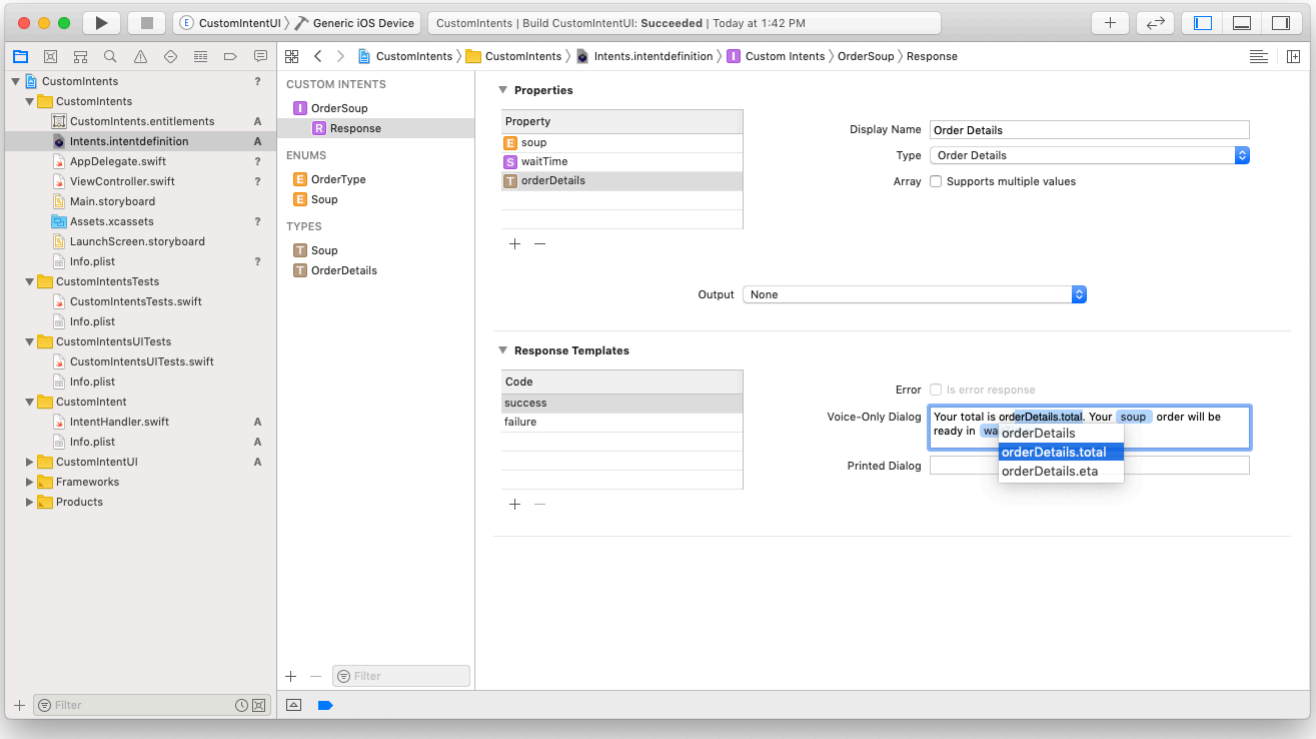


Set Intent Responses and Outputs

An intent response represents the result of the intent's action, such as ordering soup. Response templates allow Siri to respond with an appropriate dialog when a success or failure occurs. Add properties to allow passing of data into Siri dialogs or into the shortcut output. The output allows the user to use the result of your shortcut with other actions in the Shortcuts editor.

The voice-only dialog is spoken instead of the printed dialog in circumstances where the user isn't looking at a screen, such as on HomePod, CarPlay, and AirPods.

The following example shows the response for a successful order. It provides the total amount, the user's order, and the estimated wait time.



See Also

Articles

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.

- 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.