

## Article

# Generating a List of Ride Options

Generate ride options for Maps to display to the user.

## Overview

When the user selects Ride as the transportation option, Maps sends an INListRideOptions Intent object to any Intents app extensions that support the intent. In your response to that intent, provide one or more INRideOption objects representing the rides you're able to offer. Keep your list of rides reasonably short and representative of the types of vehicles and price points that are available. Don't return an exhaustive list of all vehicles available to pick up the user.

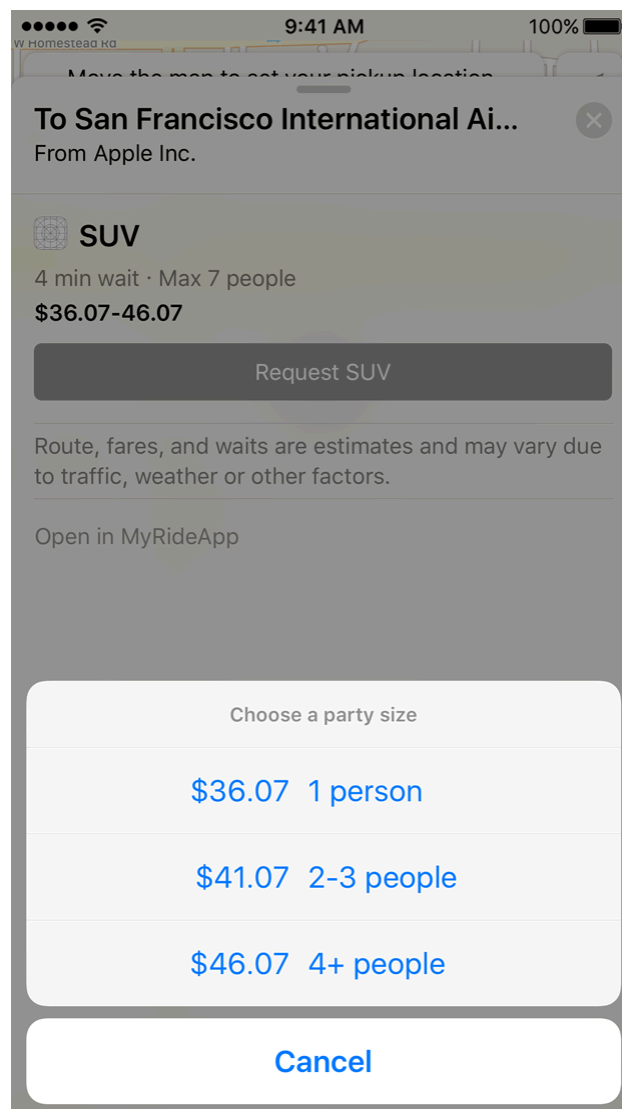
## Include Essential Information

For each ride option, include as much information as possible about the ride, including its pricing, passenger capacity, disclaimers, and an estimated time at which a vehicle of that type could arrive at the user's location. Make the information as accurate as possible, but understand that you can update the information when actually booking the ride. Here are some tips to creating your INRideOption objects:

- **Give each ride option a unique, localized name.** Ride option names should convey information about the ride type and be unique so that you can identify them later. When the user selects a ride option for booking, SiriKit passes only the option name back to you. So you must be able to identify the ride type from that string.
- **Configure the party size options when party size determines pricing.** If the number of people receiving a ride determines the price, configure the availablePartySizeOptions and availablePartySizeOptionsSelectionPrompt properties appropriately. Use your INRidePartySizeOption objects to specify the pricing information for parties of different sizes.

- **Use fare line items to enumerate individual costs.** When the cost of a ride isn't a single fixed value, provide fare line items so that the user understands how the app computed the fare. Line items can include base charges, per-mile charges, additional fees or tolls, and discounts.
- **Include an expiration date.** Expiration dates prevent your ride options from becoming stale or outdated. When a ride option expires, SiriKit sends another `INListRideOptionsIntent` object to your Intents extension.
- **If your app must complete the booking, provide an appropriate user activity object.** You might complete the booking in your app when special circumstances apply, such as when you need to change the precise pickup or drop-off location. Assigning an `NSUserActivity` object to the `userActivityForBookingInApplication` property of the `INRideOption` tells Maps that it must launch your app to complete the booking.

The following figure shows how Maps reveals the party size details to the user. When selecting a vehicle, Maps displays the base price range initially. When the user actually requests the vehicle, Maps prompts the user to select the number of passengers. Base the information for each price tier on the information you supply in the `INRidePartySizeOption` objects.



# Create Ride Options

The following code shows the creation of an INRideOption object for a compact car with fixed pricing options. In this case, the ride-booking service computes the price of the ride and quotes it directly. Because the ride's price and number of passengers can't change, it includes a disclaimer message to convey the vehicle capacity rather than assigning a value to the availablePartySizeOptions property.

```
func createCompactRideOption(pickup : CLPlacemark,
                             dropOff : CLPlacemark ) -> INRideOption {
    let ride = INRideOption(name: "Compact",
                             estimatedPickupDate: self.getCompactEstimatedPickupTime())
    ride.disclaimerMessage = "Vehicle can carry only 1 passenger."

    // Compacts have a fixed charge based on the distance.
    let charges = self.computeCompactCharges(pickup: pickup, dropOff: dropOff)
    ride.priceRange = INPriceRange(price: NSDecimalNumber(value : charges),
                                    currencyCode: "USD")

    return ride
}
```

The following code shows another example of how to create an INRideOption, in this case for an SUV type of vehicle. In this case, there are three separate pricing tiers according to the number of passengers. Each tier requires a separate INRidePartySizeOption object to include the pricing and information about the number of passengers. The ride option also fills in the priceRange property based on the prices of the different configurations.

```
func createSUVRideOption(pickup : CLPlacemark,
                         dropOff : CLPlacemark ) -> INRideOption {
    let ride = INRideOption(name: "SUV",
                             estimatedPickupDate: self.getSUVEstimatedPickupTime())

    // Configure party size options.

    let baseCharge = self.getBaseCharge()
    let perMileCharge = self.getPerMileCharge()
    let suvSurcharge = self.getSUVSurcharge()
    let extraPersonSurcharge = self.getExtraPersonSurcharge()

    // Compute the base prices for one person.
    let (rideDistance, tollCharges) = self.computeRouteDistanceAndTolls(pickup: pickup, dropOff: dropOff)
```

```

let distanceCost = perMileCharge * rideDistance
let minimumCost = distanceCost + baseCharge + suvSurcharge + tollCharges

// Configure the one-person party size.
let onePersonPriceRange = INPriceRange(price: NSDecimalNumber(value : minimumCost,
                                                                currencyCode: "USD"))
let onePersonParty = INRidePartySizeOption(partySizeRange: NSMakeRange(0, 1),
                                           sizeDescription: "1 person", priceRange: onePersonPriceRange)

// Configure the 2-3 person party size.
let threePersonPriceRange = INPriceRange(price: NSDecimalNumber(value : minimumCost +
                                                                extraPersonSurcharge), currencyCode: "USD")
let threePersonParty = INRidePartySizeOption(partySizeRange: NSMakeRange(2, 1),
                                           sizeDescription: "2-3 people", priceRange: threePersonPriceRange)

// Configure the 4+ person party size.
let fourPersonPriceRange = INPriceRange(price: NSDecimalNumber(value : minimumCost +
                                                                (2 * extraPersonSurcharge)), currencyCode: "USD")
let fourPersonParty = INRidePartySizeOption(partySizeRange: NSMakeRange(4, 3),
                                           sizeDescription: "4+ people", priceRange: fourPersonPriceRange)

// Configure the party size options
ride.availablePartySizeOptions = [onePersonParty, threePersonParty, fourPersonParty]
ride.availablePartySizeOptionsSelectionPrompt = "Choose a party size"
ride.priceRange = INPriceRange(firstPrice: NSDecimalNumber(value : minimumCost),
                               secondPrice: NSDecimalNumber(value : minimumCost +
                                                                (2 * extraPersonSurcharge)), currencyCode: "USD")

return ride
}

```

## 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.



## 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.



## 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.