Intents / INRequestRideIntentHandling

Protocol

INRequestRideIntentHandling

The handler interface for booking a ride for the user.

iOS 10.0+ | iPadOS 10.0+ | Mac Catalyst 13.1+ | visionOS 1.0+ | watchOS 3.2+

```
protocol INRequestRideIntentHandling : NSObjectProtocol
```

Overview

Use the methods of the `INRequestRideIntentHandling` protocol to resolve, confirm, and handle requests to book a ride using your service. Adopt this protocol in an object of your Intents extension that is capable of initiating the booking process with your service. SiriKit calls the methods of your object to resolve the parameters to your ride-related information and to book the ride upon confirmation by the user.

Ride requests have many items to resolve. You should resolve all options in some manner, but you may omit options that are not relevant to your service. For example, if your service transports cargo instead of passengers, you might ignore the party size option. For ride requests, you must resolve the following items:

- Pickup location
- Drop-off location
- Party size
- Ride option
- Payment method

Note

Maps does not require you to confirm the contents of a get ride status intent before handling it. User interactions drive the selection of data in Maps, ensuring that the data Maps places into an intent object is already valid.

Handling a Ride Request

Handling a ride request involves booking the ride with your service and providing SiriKit with the status of the booking. The details of how you book rides through your service are up to you. You need to schedule the ride, find a vehicle and a driver, and arrange payment from the user. The `INRequestRideIntent` object contains information about what the user needs. Pass the information in that object to your service and use the results to configure the response object you return to SiriKit.

The code listing below is an example of how you might book a ride and provide a response. Because booking the actual ride is a custom process, the example calls several custom methods to implement key phases of that process. The first of these methods uses information from the intent object to register the request with a fictional service and retrieve an ID string identifying the request. Use that ID string to retrieve specific information from the service regarding the vehicle type, driver, and ride option. Use all of the retrieved information to build the `INRideStatus` object that and add the status object to the `INRequestRideIntentResponse` object passed back to SiriKit.

Swift   Objective-C

```
func handle(requestRide intent: INRequestRideIntent, completion: (INRequestRideInter
    let status = INRideStatus()

    // Configure the ride request internally and get its ID
    status.rideIdentifier = self.createNewRideRequest(withStartingLocation: intent.p
                                                      endingLocation: intent.dropOff
                                                      partySize: intent.partySize!,
                                                      paymentMethod: intent.paymentM

    // Configure the pickup and dropoff information.
    status.estimatedPickupDate = self.estimatedPickupDateForRideRequest(identifier:
    status.pickupLocation = intent.pickupLocation
    status.dropOffLocation = intent.dropOffLocation

    // Retrieve the ride details that the user needs.
    status.vehicle = self.vehicleForRideRequest(identifier: status.rideIdentifier!)
    status.driver = self.driverForRideRequest(identifier: status.rideIdentifier!)

    // Configure the vehicle type and pricing.
    status.rideOption = self.rideOptionForRideRequest(identifier: status.rideIdenti

    // Commit the request and get the current status.
    status.phase = self.completeBookingForRideRequest(identifier: status.rideIdenti

    var responseCode : INRequestRideIntentResponseCode
    if status.phase == .received {
        responseCode = .inProgress
    }
    else if status.phase == .confirmed {
        responseCode = .success
    }
    else {
        responseCode = .failure
    }

    let response = INRequestRideIntentResponse.init(code: responseCode, userActivity

    response.rideStatus = status

    completion(response)
}
```

When creating responses, providing an `NSUserActivity` object is optional and necessary only when you want to include custom information for your app. If you specify `nil`, SiriKit creates a user activity object for you as needed. For more information on configuring the response object, see `INRequestRideIntentResponse`.

Topics

Resolving the Intent Parameters

`func resolvePickupLocation(for: INRequestRideIntent, with: (INPlacemarkResolutionResult) -> Void)`

Resolves the pickup location for the ride.

`func resolveScheduledPickupTime(for: INRequestRideIntent, with: (INDateComponentsRangeResolutionResult) -> Void)`

Resolves the pickup time for the ride.

```
func resolveDropOffLocation(for: INRequestRideIntent, with: (
INPlacemarkResolutionResult) -> Void)
```

Resolves the drop-off location for the ride.

```
func resolveRideOptionName(for: INRequestRideIntent, with: (INSpeakable
StringResolutionResult) -> Void)
```

Resolves the user’s selected ride option.

```
func resolvePartySize(for: INRequestRideIntent, with: (INInteger
ResolutionResult) -> Void)
```

Resolves the number of passengers for the ride.

Confirming the Response

```
func confirm(intent: INRequestRideIntent, completion: (INRequestRide
IntentResponse) -> Void)
```

Confirms that you can book the ride.

Handling the Intent

```
func handle(intent: INRequestRideIntent, completion: (INRequestRide
IntentResponse) -> Void)
```

Handles the booking of the ride.

Required

Relationships

Inherits From

`NSObjectProtocol`

Inherited By

`INRidesharingDomainHandling`

See Also

Request a Ride

`class INRequestRideIntent`

A request to book the specified ride from your service.

`class INRequestRideIntentResponse`

Your app’s response to a request ride intent.