Documentation

Language: **Swift**

Intents / INGetRideStatusIntentHandling

Protocol

INGetRideStatusIntentHandling

The handler interface for fetching the status of the user’s current ride.

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

```
protocol INGetRideStatusIntentHandling : NSObjectProtocol
```

Mentioned in

Providing Live Status Updates

Overview

Use the methods of the `INGetRideStatusIntentHandling` protocol to confirm and handle requests to get the status of the user’s currently booked ride. Adopt this protocol in an object of your Intents extension that is capable of retrieving the ride details from your service.

SiriKit may deliver an `INGetRideStatusIntent` object to your handler object at any time, including when the user asks explicitly for the status of that ride. The specified intent object contains no additional information. Your app must know which ride is currently active. Use the methods of this protocol to handle the request and provide an appropriate response.

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.

Updating the Ride Status Periodically

After returning your response, SiriKit may call the `startSendingUpdates(for:to:)` method to ask for further updates as the ride status changes. Implement that method and use it to begin the delivery of regular updates. Continue providing updates until SiriKit calls the `stopSendingUpdates(for:)` method.

After it starts observing the status of a ride, SiriKit keeps your Intents extension alive so that it can provide frequent updates. Always deliver an update at least once every 20 seconds; otherwise, provide updates with the same frequency that you use for updating your own app. Always provide updates when the ride phase or completion status changes. For example, provide an update when your vehicle is approaching the pickup location or arrives at it. SiriKit uses the updates to refresh any map interfaces displaying the location of your vehicle to the user.

Use the methods of the observer object to deliver your updates back to SiriKit. For more information about providing updates to the observer, see `INGetRideStatusIntentResponseObserver`.

Topics

Confirming the Response

```
func confirm(intent: INGetRideStatusIntent, completion: (INGetRideStatusIntentResponse) -> Void)
```

Confirms the details of the user’s request.

Handling the Intent

```
func handle(intent: INGetRideStatusIntent, completion: (INGetRideStatusIntentResponse) -> Void)
```

Handles providing the current status of the user’s ride.

Required

Providing Ride Status Updates

```
func startSendingUpdates(for: INGetRideStatusIntent, to: any INGetRideStatusIntentResponseObserver)
```

Start delivering regular updates about the specified ride.

Required

```
func stopSendingUpdates(for: INGetRideStatusIntent)
```

Stops the delivery of status updates for the specified ride.

Required

Relationships

Inherits From

`NSObjectProtocol`

Inherited By

`INRidesharingDomainHandling`

See Also

Get the Ride Status

`class INGetRideStatusIntent`

A request for the current status of a previously booked ride.

`class INGetRideStatusIntentResponse`

Your app’s response to a get ride status intent.

`protocol INGetRideStatusIntentResponseObserver`

An interface for providing updated status for an active ride.