

[App Clips](#) / Enabling notifications in App Clips

Article

Enabling notifications in App Clips

Enable your App Clip to schedule and receive notifications for a short or extended time period.

Overview

Some App Clips may need to schedule or receive notifications to provide value. Consider an App Clip that allows users to order food for delivery: By sending notifications, the App Clip informs the user about an upcoming delivery. If notifications are important for the functionality provided by your App Clip, enable it to schedule or receive notifications for up to 8 hours after each launch. Additionally, if you create an App Clip for multiple businesses, be sure to make changes to your notification payloads.

Important

If a user returns to a previously launched App Clip from a notification, the App Clip launches without an invocation URL. You must handle this scenario in both your App Clip and your full app. For more information on invocations, see [Responding to invocations](#).

Schedule or receive notifications temporarily

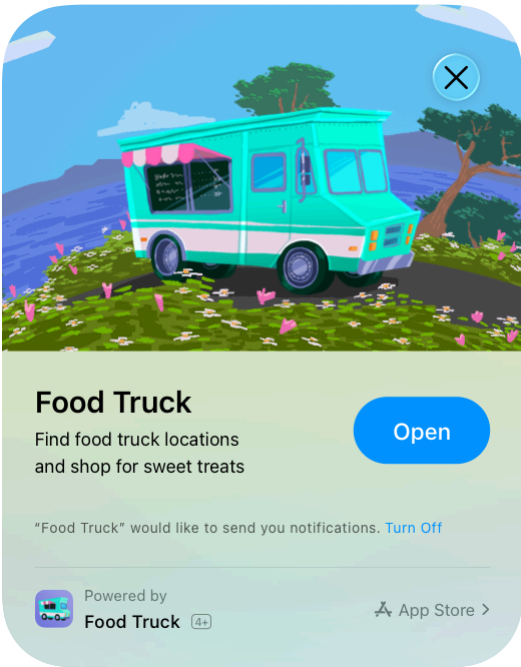
To enable your App Clip to schedule or receive notifications for up to 8 hours after each launch, first add the `NSAppClip` key to your App Clip's `Info.plist` file and set its type to `Dictionary`. Then, add an entry to the dictionary with the `NSAppClipRequestEphemeralUserNotification` key. Set its type to `Boolean` and its value to `true`.

Alternatively, open the `Info.plist` file in the property list editor and add the entry by selecting App Clip from the list of keys. This adds the `NSAppClip` key and two entries of type `Boolean` to its dictionary: "Requests ephemeral user notifications" and "Requests location confirmation". By

default, the value for both entries is NO. Change the value for “Requests ephemeral user notifications” to YES.

Info-Clip.plist		
Info-Clip > No Selection		
Key	Type	Value
Information Property List	Dictionary (4 items)	
App Clip	Dictionary (2 items)	
Requests ephemeral user notifications	Boolean	YES
Requests location confirmation	Boolean	YES
Launch Screen	Dictionary (0 items)	
UISupportedInterfaceOrientations~iPhone	Array (3 items)	
Item 0	String	UIInterfaceOrientationPortrait
Item 1	String	UIInterfaceOrientationLandscapeLeft
Item 2	String	UIInterfaceOrientationLandscapeRight
UISupportedInterfaceOrientations~iPad	Array (4 items)	
Item 0	String	UIInterfaceOrientationPortrait
Item 1	String	UIInterfaceOrientationPortraitUpsideDown
Item 2	String	UIInterfaceOrientationLandscapeLeft
Item 3	String	UIInterfaceOrientationLandscapeRight

As a result, the App Clip card that’s displayed upon invocation of the App Clip contains a note that tells the user about the App Clip’s ability to receive or schedule notifications. This permission is enabled by default, but users can disable it by tapping the note on the App Clip card.



Because users can disable notifications in the App Clip card, add code to check whether the App Clip has permission to schedule and receive notifications. The following code checks whether the

user has granted permission to send notifications for a short amount of time:

```
let center = UNUserNotificationCenter.current()
center.getNotificationSettings { (settings) in {
    if settings.authorizationStatus == .ephemeral {

        // The user didn't disable notifications in the App Clip card.
        // Add code for scheduling or receiving notifications here.
        return
    }
}
```

Request explicit permission to send notifications

If your App Clip's functionality spans more than a day, explicitly request the user's permission to send notifications. For example, a car rental company's App Clip can ask for permission to receive notifications that remind users when they need to return a rented car.

However, carefully consider whether you should ask for this permission. Users could deny the request, overriding the App Clip's ability to receive and schedule notifications for up to 8 hours after each launch.

For more information, see [Asking permission to use notifications](#).

Make changes to your notification payload

You may create an App Clip for multiple businesses. For example, you may be a platform provider for restaurants and create an App Clip that serves many different restaurants. If a user launches the App Clip consecutively for several different businesses within a short amount of time, multiple instances of the App Clip may exist on their device.

In this case, when it receives a notification, the system needs to route the notification to the appropriate App Clip instance. As a result, the system requires notification payloads to contain a URL as the target content identifier. The following code shows the notification payload for an App Clip that serves multiple businesses:

```
{
  "aps" : {
    "alert" : {
      "title" : "Order Status",
      "subtitle" : "Restaurant A",
      "body" : "Your order is ready."
```

```
    },  
    "category" : "order_status",  
    "target-content-id" : "https://example.com/restaurants/restaurant_a/order/1234"  
  }  
}
```

The value for the `target-content-id` must be a URL that matches a corresponding advanced App Clip experience. For the restaurant example, you'd register both URLs in [App Store Connect](#):

- `https://example.com/restaurants/restaurant_a`
- `https://example.com/restaurants/restaurant_b`

The invocation URLs and target content identifiers could then be:

- `https://example.com/restaurants/restaurant_a/order/1234`
- `https://example.com/restaurants/restaurant_b/order/5678`

In general, use a target content identifier that's as specific as possible. Similarly, if you enable your App Clip to schedule local notifications, set the target content identifier for the notification payload; for example, using `targetContentIdentifier`.

For more information, see [Configuring App Clip experiences](#), [Generating a remote notification](#), and [Scheduling a notification locally from your app](#).