

[App Clips](#) / Launching another app's App Clip from your app

Article

# Launching another app's App Clip from your app

Enable people to launch another app's App Clip from your app with App Clip links and offer a rich preview of it with the [Link Presentation framework](#).

## Overview

Starting with iOS 17, an app can launch another app's App Clip using the invocation URL of the App Clip. For example, if you develop several apps, your apps can launch your other apps' App Clips to allow people to use their functionality without requiring an app installation. Or, your app could offer to launch another developer's App Clip if your app offers workflows that involve usage of the other app. Depending on the App Clip and its invocation URL, choose from the following options to allow people to invoke an App Clip from your app:

- To display a rich preview and allow people to launch default or advanced App Clip experiences, use the [Link Presentation framework](#).
- To allow people to launch a default App Clip experience that makes use of the autogenerated default App Clip link, use [Link](#) or [open\(:options:completionHandler:\)](#).

## Display a preview of the App Clip

Use the [Link Presentation framework](#) to include a rich preview of the App Clip in your app that people tap to launch the App Clip directly:

1. To fetch metadata using the invocation URL of the App Clip, use [LPMetadataProvider](#).
2. To display the App Clip preview, use the metadata you receive in an [LPLinkView](#).

The following example fetches the metadata and then passes it to a link view.

```
let provider = LPMetadataProvider()
let url = URL(string: "https://appclip.example.com/")!
var linkView = ... // A reference to the link view. This could also be a custom view

provider.startFetchingMetadata(for: url) { (metadata, error) in
    guard let metadata = metadata else {
        // The fetch failed; handle the error.
        return
    }

    DispatchQueue.main.async {
        // Create the LPLinkView using the metadata. If you use a custom view, you can
        linkView = LPLinkView(metadata: metadata)
    }
}
```

## Display a direct link to an App Clip

If the App Clip uses the default App Clip link for its default App Clip experience, you can display a direct link that launches the App Clip. In a [SwiftUI](#) app, use [Link](#) as shown in the following example that displays a link to the [Backyard Birds: Building an app with SwiftData and widgets](#) app:

```
var body: some View {
    let appClipURL = URL(
        string: "https://appclip.apple.com/id?p=com.example.naturelab.backyardbirds."
    )!

    Link("Backyard Birds", destination: appClipURL)
}
```

If you use [UIKit](#), use [`open\(\_ :options:completionHandler:\)`](#) to allow people to invoke the App Clip:

```
func launchAppClip() {
    let appClipURL = URL(
        string: "https://appclip.apple.com/id?p=com.example.naturelab.backyardbirds."
    )!

    UIApplication.shared.open(appClipURL)
```

# See Also

## Launch

### Responding to invocations

Add code to respond to invocations and offer a focused launch experience.

### Associating your App Clip with your website

Enable the system to verify your App Clip to support invocations from your website and devices running iOS 16.3 or earlier.

### Supporting invocations from your website and the Messages app

Display a Smart App Banner and the App Clip card on your website that people tap to launch your App Clip, and add support for invocations from the Messages app.

### Confirming a person's physical location

Add code to quickly confirm a person's physical location while respecting their privacy.

### `APActivationPayload`

Information that's passed to an App Clip on launch.

### `NSAppClip`

A collection of keys that an App Clip uses to get additional capabilities.