

[AdAttributionKit](#) / Presenting ads in your app

Article

# Presenting ads in your app

Render different ad styles in your app.



## Overview

AdAttributionKit and StoreKit provide several ways to display in-app ads so you can customize ad display based on your app's ad positioning or other advertising goals.

The first step to presenting ads in your app is to initialize an [AppImpression](#) with an impression in compact JSON Web Signature (JWS) format. For more information about generating JWS impressions, see [Generating JWS impressions](#)

## Record view-through impressions using custom rendered ads

Custom rendered ads include ad content that overlays the app view. To record a view-through impression use the AdAttributionKit [beginView\(\)](#) and [endView\(\)](#) methods as shown in the following SwiftUI example using the `onAppear()` and `onDisappear()` view modifiers:

### Note

In UIKit, use the `viewDidAppear(_ :)` and `viewWillDisappear(_ :)` methods. Don't use or rely on timer methods to determine when to end an ad impression view in either SwiftUI or UIKit.

```
struct AdContentView: View {  
    let impression: AppImpression
```

```

var body: some View {
    VStack {
        // Advertisement content
    }
    .onAppear(perform: { handleAdAppeared() })
    .onDisappear(perform: { handleAdDisappeared() })
    .onTapGesture(perform: { handleAdTapped() })
}

init(impression: AppImpression) {
    self.impression = impression
}

func handleAdAppeared() {
    Task {
        do {
            try await impression.beginView()
        }
        catch {
            print("Failed to begin view through impression: \(error).")
        }
    }
}

func handleAdDisappeared() {
    Task {
        do {
            try await impression.endView()
        }
        catch {
            print("Failed to end view through impression: \(error).")
        }
    }
}

```

## Record click-through impressions and redirect a person to open or install the advertised app

To respond to a click-through interaction, first display a [UIEventAttributionView](#) over your ad content. Once the ad receives a tap, call [handleTap\(\)](#). The system then records a click-through impression, and if the app specified by the impression's advertised item ID isn't installed,

the system launches the app's product page on the App Store or alternative marketplace according to the user's preferences in Settings. If the app is already installed, the system launches the app directly.

```
func handleAdTapped(impression: AppImpression) async {
    do {
        // This fails if a person didn't tap `UIEventAttributionView`.
        try await impression.handleTap()
    }
    catch {
        print("Failed to perform click through impression: \(error).")
    }
}
```

## Display StoreKit rendered ads

Pass an `AppImpression` when configuring a StoreKit rendered ad, and it handles recording a view-through impression after the framework displays it for 2 seconds; it records a click-through impression if a person taps through the ad. For more information on StoreKit rendered ads, see [SKStoreProductViewController](#), [SKOverlay.AppConfiguration](#), [appImpression](#), and [loadProduct\(parameters:impression:\)](#)

## See Also

### Essentials

- 📄 Understanding AdAttributionKit and SKAdNetwork interoperability
  - Learn how attribution APIs interact to deliver ad impressions.
- 📄 Receiving ad attributions and postbacks
  - Understand timeframes and priorities for ad impressions that result in ad attributions, and how impressions qualify for postbacks.
- 📄 Identifying conversion values with conversion tags
  - Use conversion tags to identify and update specific postbacks when you have overlapping conversion windows.