

Making your app content shareable

Add support for universal links and a Shared with You shelf to support shared content in your app.



Overview

System apps like Safari, News, Music, Photos, and Podcasts use Shared with You to make it easier to share content with friends and family. By adopting universal links and adding a Shared with You shelf to your app, you can also share your content using Messages. When you add views to showcase shared content activity, your app allows people to view Messages conversations that contain your shared content and still stay focused on your app.

Your app can also manage who can view or share your content using the [SWAttributionView](#). This adds a new level of collaboration to your app. You can customize your content's contextual menu to add even more functionality to shared content.

Adopt universal links

Your app uses universal links to share content with other apps. When a user activates a universal link, the system launches your app and sends it an [NSUserActivity](#) object. Query this object to take actions for your shared content.

To support universal links in your app:

1. Create a two-way association between your app and your website and specify the URLs that your app handles. See [Supporting associated domains](#).
2. Update your app delegate to respond when it receives an [NSUserActivity](#) object with the [activityType](#) set to [NSUserActivityTypeBrowsingWeb](#).

For more information, see [Supporting universal links in your app](#).

Add Shared with You capability

To add the Shared with You capability in Xcode, follow these steps:

1. Select your project target.
2. Select the Signing & Capabilities pane.
3. Click + Capability to bring up the [Capabilities](#) library.
4. Choose the Shared with You capability.

For more information, see [Adding capabilities to your app](#).

Add Shared with You shelf

Instead of relying on Messages to display shared content activity, you can add a Shared with You shelf to your app. This is a custom view your app creates and manages to display [SWHighlightCenter](#) activity. People can see new content shares and interact with existing ones from your app.

```
// Enumerate Shared with You shelf.

class SharedWithYouViewController: UIViewController, SWHighlightCenterDelegate {
    let highlightCenter = SWHighlightCenter()

    override func viewDidLoad() {
        super.viewDidLoad()
        highlightCenter.delegate = self
    }

    func highlightCenterHighlightsDidChange(_ highlightCenter: SWHighlightCenter) {
        for highlight in highlightCenter.highlights {
            let highlightURL = highlight.url
            // Generate a rich preview for this highlight.
        }
    }
}
```

Add an SWAttributionView to your content

The [SWAttributionView](#) is the user interface your app customizes to give people a way to interact with [SWHighlight](#) events, like the addition of a new participant or a title change to the shared content. You can adjust the alignment of the content, the width, and the [SWAttributionView.DisplayContext](#) format.

```
// Set the horizontal alignment for an attribution view.

let attributionView = SWAttributionView()
attributionView.highlight = self.highlightCenter.highlights[index]
attributionView.preferredMaxLayoutWidth = maximumWidthForView
attributionView.horizontalAlignment = .leading
```

You can show the displayContext as an [SWAttributionView.DisplayContext.summary](#) or an [SWAttributionView.DisplayContext.detail](#) view. The detail view is ideal if you have more vertical space in your app to provide a larger image of the shared content and more related information.

```
// Set the display context for an attribution view.

let attributionView = SWAttributionView()
attributionView.highlight = self.highlightCenter.highlights[index]
attributionView.preferredMaxLayoutWidth = maximumWidthForView
attributionView.horizontalAlignment = .center
attributionView.displayContext = .summary
```

You can also customize the highlight menu text and add additional menu items.

```
// Add Shared with You content menu to your app's content.

let attributionView = SWAttributionView()
attributionView.highlight = self.highlightCenter.highlights[index]
attributionView.menuTitleForHideAction = "Remove Item"

let contextMenuConfig = UIContextMenuConfiguration(identifier: nil, previewProvider: nil) { [weak self]
    let additionalMenu = attributionView.supplementalMenu
    // Append additional menu items to your content menu.
}
```

See Also

Shared content

☰ Shared content interactions

Use highlights and attribution views to manage participants and trigger events for shared content.

Platforms	Topics & Technologies	Resources	Programs
iOS	Accessibility	Documentation	Apple Developer Program
iPadOS	Accessories	Tutorials	Apple Developer Enterprise Program
macOS	App Extension	Downloads	App Store Small Business Program
tvOS	App Store	Forums	MFi Program
visionOS	Audio & Video	Videos	News Partner Program
watchOS	Augmented Reality	Support	Video Partner Program
Tools	Design	Support Articles	Security Bounty Program
Swift	Distribution	Contact Us	Security Research Device Program
SwiftUI	Education	Bug Reporting	Events
Swift Playground	Fonts	System Status	Meet with Apple
TestFlight	Games	Account	Apple Developer Centers
Xcode	Health & Fitness	Apple Developer	App Store Awards
Xcode Cloud	In-App Purchase	App Store Connect	Apple Design Awards
SF Symbols	Localization	Certificates, IDs, & Profiles	Apple Developer Academies
	Maps & Location	Feedback Assistant	WWDC
	Machine Learning & AI		
	Open Source		
	Security		
	Safari & Web		