

[SwiftUI](#) / [Landmarks: Building an app with Liquid Glass](#) / Landmarks: Applying a background extension effect

Sample Code

Landmarks: Applying a background extension effect

Configure an image to blur and extend under a sidebar or inspector panel.

[Download](#)

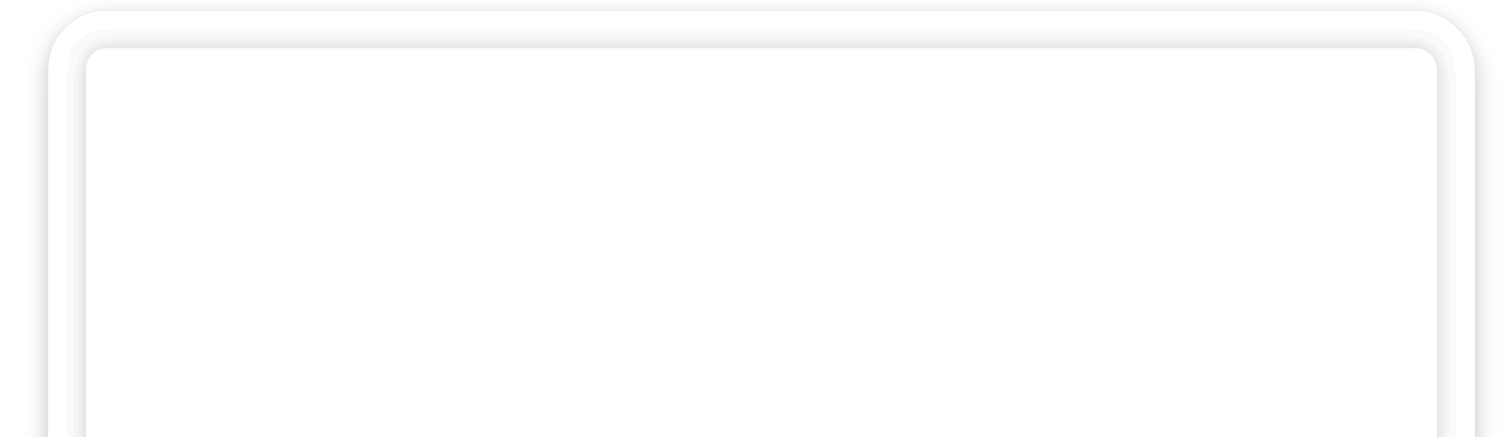
iOS 26.0+ | iPadOS 26.0+ | macOS 26.0+ | Xcode 26.0+

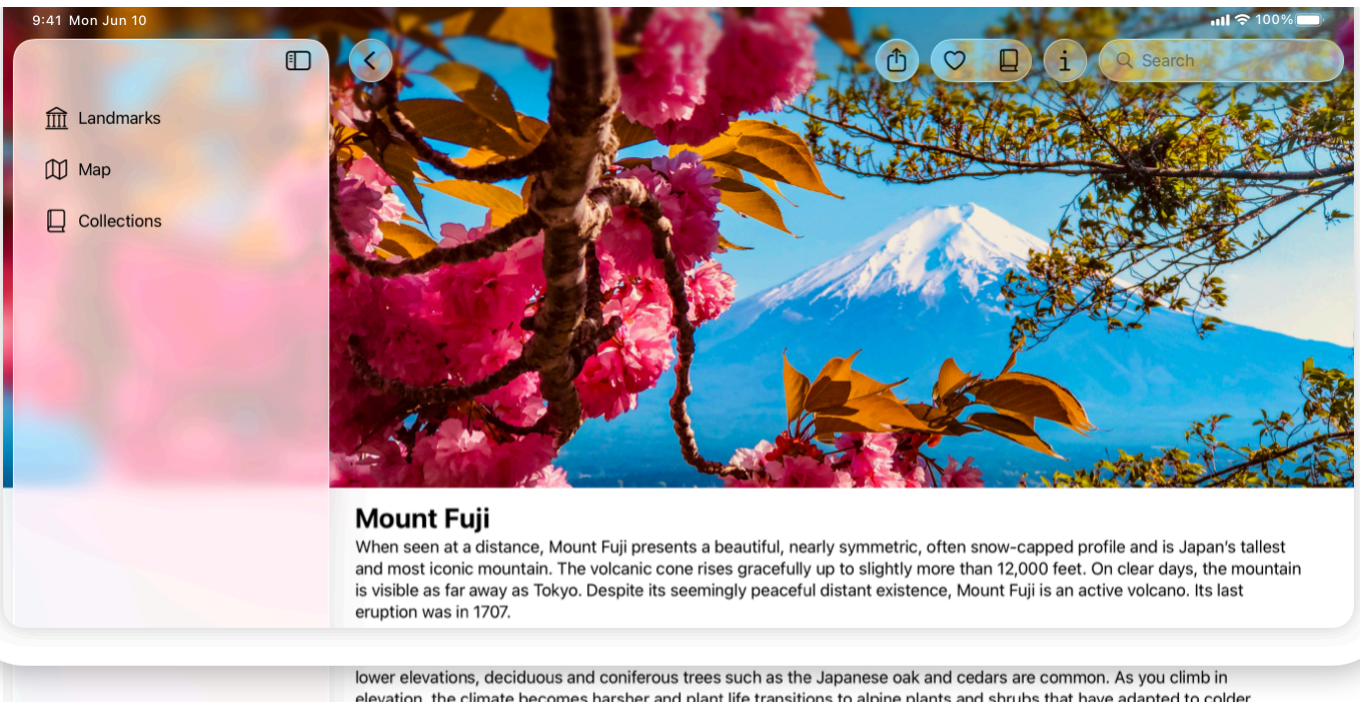
Overview

The Landmarks app lets people explore interesting sites around the world. Whether it's a national park near their home or a far-flung location on a different continent, the app provides a way for people to organize and mark their adventures and receive custom activity badges along the way.

This sample demonstrates how to apply a background extension effect. In the top Landmarks view, the sample applies a background extension effect to the featured image in `LandmarksView`, and to the main image in `LandmarkDetailView`. The background extension effect blurs and extends the image under the sidebar or inspector panel when open. The following images show the main image in `LandmarkDetailView` both with and without the background extension effect.

With Without





To apply the background extension effect, the sample:

1. Aligns the view to the leading and trailing edges of the containing view.
2. Applies the `backgroundExtensionEffect()` modifier to the view.
3. Configures only the image in the background extension, and avoids applying the effect to the title and button in the overlay.

Align the view to the leading and trailing edges

To apply the `backgroundExtensionEffect()` to a view, align the leading edge of the view next to the sidebar, and align the trailing edge of the view to the trailing edge of the containing view.

In `LandmarksView`, the `LandmarkFeaturedItemView` and the containing `LazyVStack` and `ScrollView` don't have padding. This allows the `LandmarkFeaturedItemView` to align with the leading edge of the view next to the sidebar.

```
ScrollView(showsIndicators: false) {
    LazyVStack(alignment: .leading, spacing: Constants.standardPadding) {
        LandmarkFeaturedItemView(landmark: modelData.featuredLandmark!)
            .flexibleHeaderContent()
        //...
    }
}
```

In `LandmarkDetailView`, the `ScrollView` and `VStack` that contain the main image also don't have any padding. This allows the main image to align against the leading edge of the containing view.

Apply the background extension effect to the image

In `LandmarkDetailView`, the sample applies the background extension effect to the main image by adding the `backgroundExtensionEffect()` modifier:

```
Image(landmark.backgroundImageName)
    //...
    .backgroundExtensionEffect()
```

When the sidebar is open, the system extends the image in the leading direction as follows:

- The system takes a section of the leading end of the image that matches the width of the sidebar.
- The system flips that portion of the image horizontally toward the leading edge and applies a blur to the flipped section.
- The system places the modified section of the image under the sidebar, immediately before the leading edge of the image.

When the inspector is open, the system extends the image in the trailing direction as follows:

- The system takes a section of the trailing end of the image that matches the width of the sidebar.
- The system flips that portion of the image horizontally toward the trailing edge and applies a blur to the flipped section.
- The system places the modified section of the image under the inspector, immediately after the trailing edge of the image.

Configure only the image

In `LandmarksView`, the `LandmarkFeaturedItemView` has an image from the featured landmark, and includes a title for the landmark and a button you can click or tap to learn more about that location.

To avoid having the landmark's title and button appear under the sidebar in macOS, the sample applies the `backgroundExtensionEffect()` modifier to the image before adding the overlay that includes the title and button:

```

Image(decorative: landmark.backgroundImageName)
//...
.backgroundExtensionEffect()
.overlay(alignment: .bottom) {
    VStack {
        Text("Featured Landmark", comment: "Big headline in the main image of featured landmark")
        //...
        Text(landmark.name)
        //...
        Button("Learn More") {
            modelData.path.append(landmark)
        }
        //...
    }
    .padding([.bottom], Constants.learnMoreBottomPadding)
}

```

See Also

App features

- {} Landmarks: Extending horizontal scrolling under a sidebar or inspector
Improve your horizontal scrollbar's appearance by extending it under a sidebar or inspector.
- {} Landmarks: Refining the system provided Liquid Glass effect in toolbars
Organize toolbars into related groupings to improve their appearance and utility.
- {} Landmarks: Displaying custom activity badges
Provide people with a way to mark their adventures by displaying animated custom activity badges.