

[WidgetKit](#) / Building Widgets Using WidgetKit and SwiftUI

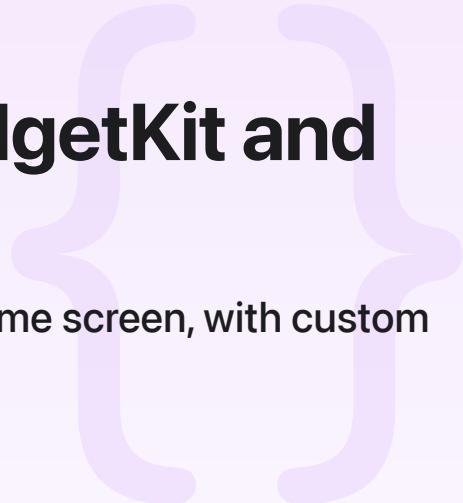
Sample Code

Building Widgets Using WidgetKit and SwiftUI

Create widgets to show your app's content on the Home screen, with custom intents for user-customizable settings.

[Download](#)

iOS 14.0+ | iPadOS 14.0+ | Xcode 13.2+ | Mac Catalyst 14.0+



Overview

Note

This sample code project is associated with WWDC20 session [10034: Widgets Code-Along, Part 1: The Adventure Begins](#); session [10035: Widgets Code-Along, Part 2: Alternate Timelines](#); and session [10036: Widgets Code-Along, Part 3: Advancing Timelines](#).

See Also

Widget creation

-  Creating a widget extension

Display your app's content in a convenient, informative widget on various devices.
-  Supporting additional widget sizes

Offer widgets in additional contexts by adding support for various widget sizes.

- 📄 Creating accessory widgets and watch complications
 - Support accessory widgets that appear on the Lock Screen and as complications on Apple Watch.
- 📄 Migrating ClockKit complications to WidgetKit
 - Leverage WidgetKit's API to create watchOS complications using SwiftUI.
- {} Emoji Rangers: Supporting Live Activities, interactivity, and animations
 - Offer Live Activities, controls, animate data updates, and add interactivity to widgets.

`struct StaticConfiguration`

An object describing the content of a widget that has no user-configurable options.

`enum WidgetFamily`

Values that define the widget's size and shape.

`struct WidgetRenderingMode`

Constants that indicate the rendering mode for a widget.