

[SwiftUI / WatchKit integration](#)

API Collection

# WatchKit integration

Add WatchKit views to your SwiftUI app, or use SwiftUI views in your WatchKit app.

## Overview

Integrate SwiftUI with your app's existing content using hosting controllers to add SwiftUI views into WatchKit interfaces. A hosting controller wraps a set of SwiftUI views in a form that you can then add to your storyboard-based app.



You can also add WatchKit views and view controllers to your SwiftUI interfaces. A representable object wraps the designated view or view controller, and facilitates communication between the wrapped object and your SwiftUI views.

For design guidance, see [Designing for watchOS](#) in the Human Interface Guidelines.

## Topics

# Displaying SwiftUI views in WatchKit

```
class WKHostingController
```

A WatchKit interface controller that hosts a SwiftUI view hierarchy.

```
class WKUserNotificationHostingController
```

A WatchKit user notification interface controller that hosts a SwiftUI view hierarchy.

## Adding WatchKit views to SwiftUI view hierarchies

```
protocol WKInterfaceObjectRepresentable
```

A view that represents a WatchKit interface object.

```
struct WKInterfaceObjectRepresentableContext
```

Contextual information about the state of the system that you use to create and update your WatchKit interface object.

---

## See Also

### Framework integration

#### ☰ AppKit integration

Add AppKit views to your SwiftUI app, or use SwiftUI views in your AppKit app.

#### ☰ UIKit integration

Add UIKit views to your SwiftUI app, or use SwiftUI views in your UIKit app.

#### ☰ Technology-specific views

Use SwiftUI views that other Apple frameworks provide.