

[SwiftUI](#) / [View](#) / `onContinueUserActivity(_:perform:)`

## Instance Method

# onContinueUserActivity(\_:perform:)

Registers a handler to invoke in response to a user activity that your app receives.

iOS 14.0+ | iPadOS 14.0+ | Mac Catalyst 14.0+ | macOS 11.0+ | tvOS 14.0+ | visionOS 1.0+ | watchOS 7.0+

```
nonisolated
func onContinueUserActivity(
    _ activityType: String,
    perform action: @escaping (NSUserActivity) -> ()
) -> some View
```

## Parameters

### activityType

The type of activity that the `action` closure handles. Be sure that this string matches one of the values that you list in the [NSUserActivityTypes](#) array in your app's Information Property List.

### action

A closure that SwiftUI calls when your app receives a user activity of the specified type. The closure takes the activity as an input parameter.

## Return Value

A view that handles incoming user activities.

## Discussion

Use this view modifier to receive `NSUserActivity` instances in a particular scene within your app. The scene that SwiftUI routes the incoming user activity to depends on the structure of your app, what scenes are active, and other configuration. For more information, see [handlesExternalEvents\(matching:\)](#).

UI frameworks traditionally pass Universal Links to your app using a user activity. However, SwiftUI passes a Universal Link to your app directly as a URL. To receive a Universal Link, use the [onOpenURL\(perform:\)](#) modifier instead.

---

## See Also

### Sending and receiving user activities

`{}` Restoring your app's state with SwiftUI

Provide app continuity for users by preserving their current activities.

```
func userActivity<P>(String, element: P?, (P, NSUserActivity) -> ()) ->  
some View
```

Advertises a user activity type.

```
func userActivity(String, isActive: Bool, (NSUserActivity) -> ()) ->  
some View
```

Advertises a user activity type.