Documentation Language: Swift

ActivityKit / Activity

Class

Activity

The object you use to start, update, and end a Live Activity.

iOS 16.1+ | iPadOS 16.1+

```
class Activity<Attributes> where Attributes : ActivityAttributes
```

Mentioned in

Displaying live data with Live Activities

Overview

The `Activity` object offers functionality to start, update, and end a Live Activity from within your app. You can update or end a Live Activity while your app is in the background, but you can only start a Live Activity while the app is in the foreground, unless you adopt App Intents and start the Live Activity using a `LiveActivityIntent`.

Additionally, `Activity` offers functionality to observe changes to:

- The Live Activity
- The Live Activity’s state in its life cycle
- A person’s permission to start Live Activities
- The Live Activity’s push token if you configure it to receive updates through ActivityKit push notifications.

To observe these changes, use the asynchronous sequences the activity object offers; for example, use the `activityStateUpdates` sequence to observe changes to the state of a Live Activity.

Topics

Starting a Live Activity

```
static func request(attributes: Attributes, content: ActivityContent<
Activity<Attributes>.ContentState>, pushType: PushType?) throws ->
Activity<Attributes>
```

Requests and starts a standard Live Activity.

```
static func request(attributes: Attributes, content: ActivityContent<
Activity<Attributes>.ContentState>, pushType: PushType?, style:
ActivityStyle) throws -> Activity<Attributes>
```

Requests and starts a Live Activity.

```
static func request(attributes: Attributes, content: ActivityContent<
Activity<Attributes>.ContentState>, pushType: PushType?, style:
ActivityStyle, alertConfiguration: AlertConfiguration, start: Date)
throws -> Activity<Attributes>
```

Requests and schedules a Live Activity for a specific date.

~~static func request(attributes: Attributes, content: ActivityContent< Activity<Attributes>.ContentState>, pushType: PushType?, style: ActivityStyle, alertConfiguration: AlertConfiguration, startDate: Date) throws -> Activity<Attributes>~~

Deprecated

```
let attributes: Attributes
```

A set of attributes that describe a Live Activity and its content.

```
protocol ActivityAttributes
```

The protocol you implement to describe the content of a Live Activity.

```
enum ActivityStyle
```

```
var content: ActivityContent<Activity<Attributes>.ContentState>
```

The dynamic content of a Live Activity.

```
struct ActivityContent
```

A structure that describes the state and configuration of a Live Activity.

```
typealias ContentState
```

The type alias for the structure that describes the dynamic content of a Live Activity.

```
struct PushType
```

The structure that offers constants you use to configure a Live Activity to receive updates through ActivityKit push notifications.

```
enum ActivityAuthorizationError
```

An error that indicates why the request to start a Live Activity failed.

Updating a Live Activity

```
func update(ActivityContent<Activity<Attributes>.ContentState>) async
```

Updates the dynamic content of the Live Activity.

```
func update(ActivityContent<Activity<Attributes>.ContentState>, alertConfiguration: AlertConfiguration?) async
```

Updates the dynamic content of a Live Activity and alerts a person about the Live Activity update.

```
struct AlertConfiguration
```

A structure you use to configure an alert that appears when you update your Live Activity.

```
func update(ActivityContent<Activity<Attributes>.ContentState>, alertConfiguration: AlertConfiguration?, timestamp: Date) async
```

Updates the dynamic content of a Live Activity and alerts a person about the Live Activity update.

Ending a Live Activity

```
func end(ActivityContent<Activity<Attributes>.ContentState>?, dismissalPolicy: ActivityUIDismissalPolicy) async
```

Ends an active Live Activity.

```
struct ActivityUIDismissalPolicy
```

The structure that describes when the system should remove a Live Activity that ended.

```
func end(ActivityContent<Activity<Attributes>.ContentState>?, dismissalPolicy: ActivityUIDismissalPolicy, timestamp: Date) async
```

Ends an active Live Activity.

Observing Live Activity content changes

```
var contentUpdates: Activity<Attributes>.ContentUpdates
```

An asynchronous sequence you use to observe changes to the dynamic content of a Live Activity.

```
struct ContentUpdates
```

A structure that offers functionality to observe changes to the dynamic content of a Live Activity.

Observing the Live Activity life cycle

```
var activityState: ActivityState
```

The current state of a Live Activity in its life cycle.

```
enum ActivityState
```

The enum that describes the state of a Live Activity in its life cycle.

```
var activityStateUpdates: Activity<Attributes>.ActivityStateUpdates
```

An asynchronous sequence you use to observe activity state changes.

```
struct ActivityStateUpdates
```

A structure that offers functionality to observe state changes of a Live Activity.

Using ActivityKit push notifications

```
var pushToken: Data?
```

The token you use to send ActivityKit push notifications to a Live Activity.

```
var pushTokenUpdates: Activity<Attributes>.PushTokenUpdates
```

An asynchronous sequence you use to observe changes to the push token of a Live Activity.

```
struct PushTokenUpdates
```

A structure that offers functionality to observe changes to the push token of a Live Activity.

```
static var pushToStartToken: Data?
```

The token you use to start a Live Activity with an ActivityKit push notification.

```
static var pushToStartTokenUpdates: Activity<Attributes>.PushTokenUpdates
```

An asynchronous sequence you use to observe changes to the token for starting a Live Activity with an ActivityKit push notification.

Checking user authorization

```
class ActivityAuthorizationInfo
```

An object with information about whether a person allowed your app to start Live Activities and permitted content updates with frequent ActivityKit push notifications.

Accessing Live Activities

```
static var activities: [Activity<Attributes>]
```

An array of your app’s current Live Activities.

```
static var activityUpdates: Activity<Attributes>.ActivityUpdates
```

An asynchronous sequence you use to observe changes to ongoing Live Activities and to asynchronously access a Live Activity when you start it.

```
struct ActivityUpdates
```

A structure that offers functionality to observe changes to a Live Activity.

Identifying a Live Activity

```
let id: String
```

A unique identifier for a Live Activity.

```
let id: String
```

A unique identifier for a Live Activity.

Deprecated

Deprecated symbols

Review unsupported symbols and their replacements.

Relationships

Conforms To

`Identifiable`

See Also

Starting a Live Activity

Displaying live data with Live Activities

Display up-to-date data and offer quick interactions in the Dynamic Island, on the Lock Screen, in CarPlay, and on a paired Mac or Apple Watch.

Starting and updating Live Activities with ActivityKit push notifications

Use ActivityKit to receive push tokens and to remotely start, update, and end your Live Activity with ActivityKit notifications.

{} Emoji Rangers: Supporting Live Activities, interactivity, and animations

Offer Live Activities, controls, animate data updates, and add interactivity to widgets.

`NSSupportsLiveActivities`

A Boolean value that indicates whether an app supports Live Activities.

`NSSupportsLiveActivitiesFrequentUpdates`

A Boolean value that indicates whether an app can update its Live Activities frequently.