

[ActivityKit](#) / ActivityAttributes

Protocol




ActivityAttributes

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

iOS 16.1+ | iPadOS 16.1+

```
protocol ActivityAttributes : Decodable, Encodable
```

Mentioned in

-  [Displaying live data with Live Activities](#)
-  [Launching your app from a Live Activity](#)
-  [Starting and updating Live Activities with ActivityKit push notifications](#)

Overview

The ActivityAttributes protocol describes the content that appears in your Live Activity. Its inner type [ContentState](#) represents the dynamic content of the Live Activity.

The following example shows an implementation of the ActivityAttributes protocol for a pizza delivery app. The app's Live Activity shows the number of ordered pizzas and the total amount on the bill as static data and the name of the driver and an estimated delivery time as dynamic data that changes over time. Note how the implementation defines the type alias `PizzaDeliveryStatus` to make the code more descriptive and easier to read.

```
public import Foundation
import ActivityKit

struct PizzaDeliveryAttributes: ActivityAttributes {
    public typealias PizzaDeliveryStatus = ContentState
```

```
public struct ContentState: Codable, Hashable {  
    var driverName: String  
    var deliveryTimer: ClosedRange<Date>  
}  
  
var numberOfPizzas: Int  
var totalAmount: String  
var orderNumber: String  
}
```

Topics

Dynamic content

`associatedtype ContentState : Decodable, Encodable, Hashable`

The associated type that describes the dynamic content of a Live Activity.

Required

Instance Methods

```
func previewContext(Self.ContentState, isStale: Bool, viewKind:  
ActivityPreviewViewKind) -> some View
```

Generates a preview for a Live Activity.

Relationships

Inherits From

`Decodable, Encodable`

See Also

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.

```
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.

