Documentation Language: Swift

UIKit / UIApplicationDelegate

Protocol

UIApplicationDelegate

A set of methods to manage shared behaviors for your app.

iOS | iPadOS | Mac Catalyst | tvOS | visionOS

```
@MainActor
protocol UIApplicationDelegate : NSObjectProtocol
```

Overview

Your app delegate object manages your app's shared behaviors. The app delegate is effectively the root object of your app, and it works in conjunction with `UIApplication` to manage some interactions with the system. Like the `UIApplication` object, UIKit creates your app delegate object early in your app's launch cycle so it's always present.

Use your app delegate object to handle the following tasks:

- Initializing your app's central data structures
- Configuring your app's scenes
- Responding to notifications originating from outside the app, such as low-memory warnings, download completion notifications, and more
- Responding to events that target the app itself, and aren't specific to your app's scenes, views, or view controllers
- Registering for any required services at launch time, such as Apple Push Notification service

For more information about how you use the app delegate object to initialize your app at launch time, see Responding to the launch of your app.

Life-cycle management in iOS 12 and earlier

In iOS 12 and earlier, you use your app delegate to manage major life cycle events in your app. Specifically, you use methods of the app delegate to update the state of your app when it enters the foreground or moves to the background.

- For information on what to do when your app enters the foreground, see Preparing your UI to run in the foreground.
- For information on what to do when your app enters the background, see Preparing your UI to run in the background.
- For general information about the life cycle of your app, see Managing your app's life cycle.

Topics

Initializing the app

```
func application(UIApplication, willFinishLaunchingWithOptions: [UIApplication.LaunchOptionsKey : Any]?) -> Bool
```

Tells the delegate that the launch process has begun.

```
func application(UIApplication, didFinishLaunchingWithOptions: [UIApplication.LaunchOptionsKey : Any]?) -> Bool
```

Tells the delegate that the launch process is almost done and the app is almost ready to run.

```
struct LaunchOptionsKey
```

The keys you use to access values in the launch options dictionary that the system passes to your app at initialization.

```
class let didFinishLaunchingNotification: NSNotification.Name
```

A notification that posts immediately after the app finishes launching.

Configuring and discarding scenes

```
func application(UIApplication, configurationForConnecting: UISceneSession, options: UIScene.ConnectionOptions) -> UISceneConfiguration
```

Retrieves the configuration data for UIKit to use when creating a new scene.

```
func application(UIApplication, didDiscardSceneSessions: Set<UISceneSession>)
```

Tells the delegate that the user closed one or more of the app's scenes from the app switcher.

Responding to app life-cycle events

~~`func applicationDidBecomeActive(UIApplication)`~~

Tells the delegate that the app has become active.

Deprecated

~~`func applicationWillResignActive(UIApplication)`~~

Tells the delegate that the app is about to become inactive.

Deprecated

`func applicationDidEnterBackground(UIApplication)`

Tells the delegate that the app is now in the background.

~~`func applicationWillEnterForeground(UIApplication)`~~

Tells the delegate that the app is about to enter the foreground.

Deprecated

`func applicationWillTerminate(UIApplication)`

Tells the delegate when the app is about to terminate.

`class let didBecomeActiveNotification: NSNotification.Name`

A notification that posts when the app becomes active.

`class let didEnterBackgroundNotification: NSNotification.Name`

A notification that posts when the app enters the background.

`class let willEnterForegroundNotification: NSNotification.Name`

A notification that posts shortly before an app leaves the background state on its way to becoming the active app.

`class let willResignActiveNotification: NSNotification.Name`

A notification that posts when the app is no longer active and loses focus.

`class let willTerminateNotification: NSNotification.Name`

A notification that posts when the app is about to terminate.

Responding to environment changes

```
func applicationProtectedDataDidBecomeAvailable(UIApplication)
```

Tells the delegate that protected files are available now.

```
func applicationProtectedDataWillBecomeUnavailable(UIApplication)
```

Tells the delegate that the protected files are about to become unavailable.

```
func applicationDidReceiveMemoryWarning(UIApplication)
```

Tells the delegate when the app receives a memory warning from the system.

```
func applicationSignificantTimeChange(UIApplication)
```

Tells the delegate when there is a significant change in the time.

```
class let protectedDataDidBecomeAvailableNotification: NSNotification.Name
```

A notification that posts when the protected files become available for your code to access.

```
class let protectedDataWillBecomeUnavailableNotification: NSNotification.Name
```

A notification that posts shortly before protected files are locked down and become inaccessible.

```
class let didReceiveMemoryWarningNotification: NSNotification.Name
```

A notification that posts when the app receives a warning from the operating system about low memory availability.

```
class let significantTimeChangeNotification: NSNotification.Name
```

A notification that posts when there’s a significant change in time.

Managing app state restoration

```
func application(UIApplication, shouldSaveSecureApplicationState: NSCoder) -> Bool
```

Asks the delegate whether to securely preserve the app’s state.

```
func application(UIApplication, shouldRestoreSecureApplicationState: NSCoder) -> Bool
```

Asks the delegate whether to restore the app’s saved state.

```
func application(UIApplication, viewControllerWithRestorationIdentifierPath: [String], coder: NSCoder) -> UIViewController?
```

Asks the delegate to provide the specified view controller.

```
func application(UIApplication, willEncodeRestorableStateWith: NSCoder)
```

Tells your delegate to save any high-level state information at the beginning of the state preservation process.

```
func application(UIApplication, didDecodeRestorableStateWith: NSCoder)
```

Tells your delegate to restore any high-level state information as part of the state restoration process.

```
class let stateRestorationBundleVersionKey: String
```

The version of your app responsible for creating the restoration archive.

```
class let stateRestorationSystemVersionKey: String
```

The version of the system on which your app created the restoration archive.

```
class let stateRestorationTimestampKey: String
```

The time your app created the restoration archive.

```
class let stateRestorationUserInterfaceIdiomKey: String
```

The user interface idiom that was in effect when your app created the restoration archive.

```
class let stateRestorationViewControllerStoryboardKey: String
```

A reference to the storyboard that contains the view controller.

Downloading data in the background

```
func application(UIApplication, handleEventsForBackgroundURLSession: String, completionHandler: () -> Void)
```

Tells the delegate that events related to a URL session are waiting to be processed.

```
enum UIBackgroundFetchResult
```

Constants that indicate the result of a background fetch operation.

Handling remote notification registration

```
func application(UIApplication, didRegisterForRemoteNotificationsWithDeviceToken: Data)
```

Tells the delegate that the app successfully registered with Apple Push Notification service (APNs).

```
func application(UIApplication, didFailToRegisterForRemoteNotificationsWithError: any Error)
```

Tells the delegate when Apple Push Notification service cannot successfully complete the registration process.

`func application(UIApplication, didReceiveRemoteNotification: [AnyHashable : Any], fetchCompletionHandler: (UIBackgroundFetchResult) -> Void)`

Tells the app that a remote notification arrived that indicates there is data to be fetched.

Continuing user activity and handling quick actions

~~func application(UIApplication, willContinueUserActivityWithType: String) -> Bool~~

Tells the delegate if your app takes responsibility for notifying users when a continuation activity takes longer than expected.

Deprecated

~~func application(UIApplication, continue: NSUserActivity, restorationHandler: ([any UIUserActivityRestoring]?) -> Void) -> Bool~~

Tells the delegate that the data for continuing an activity is available.

Deprecated

~~func application(UIApplication, didUpdate: NSUserActivity)~~

Tells the delegate that the activity was updated.

Deprecated

~~func application(UIApplication, didFailToContinueUserActivityWithType: String, error: any Error)~~

Tells the delegate that the activity couldn't be continued.

Deprecated

~~func application(UIApplication, performActionFor: UIApplicationShortcutItem, completionHandler: (Bool) -> Void)~~

Tells the delegate that the user selected a Home screen quick action for your app, except when you've intercepted the interaction in a launch method.

Deprecated

Interacting with WatchKit

`func application(UIApplication, handleWatchKitExtensionRequest: [AnyHashable : Any]?, reply: ([AnyHashable : Any]?) -> Void)`

Asks the delegate to respond to a request from a paired watchOS app.

Interacting with HealthKit

```
func applicationShouldRequestHealthAuthorization(UIApplication)
```

Tells the delegate when your app should ask the user for access to his or her HealthKit data.

Opening a URL-specified resource

~~func application(UIApplication, open: URL, options: [UIApplication.OpenURLOptionsKey : Any]) -> Bool~~

Asks the delegate to open a resource specified by a URL, and provides a dictionary of launch options.

Deprecated

~~struct OpenURLOptionsKey~~

Keys you use to access values in the options dictionary when opening a URL.

Deprecated

Disallowing specified app extension types

```
func application(UIApplication, shouldAllowExtensionPointIdentifier:
UIApplication.ExtensionPointIdentifier) -> Bool
```

Asks the delegate to grant permission to use app extensions that are based on a specified extension point identifier.

```
struct ExtensionPointIdentifier
```

A structure that identifies types of extensions.

```
static let keyboard: UIApplication.ExtensionPointIdentifier
```

The identifier for custom keyboards.

Handling SiriKit intents

```
func application(UIApplication, handlerFor: INIntent) -> Any?
```

Asks the delegate for an intent handler capable of handling the specified intent.

Handling CloudKit invitations

~~func application(UIApplication, userDidAcceptCloudKitShareWith: CKShareMetadata)~~

Tells the delegate that the app now has access to shared information in CloudKit.

Deprecated

Localizing keyboard shortcuts

```
func applicationShouldAutomaticallyLocalizeKeyCommands(UIApplication) -> Bool
```

Returns a Boolean value that tells the system whether to remap menu shortcuts to support localized keyboards.

Managing interface geometry

```
func application(UIApplication, supportedInterfaceOrientationsFor: UIWindow?) -> UIInterfaceOrientationMask
```

Asks the delegate for the interface orientations to use for the view controllers in the specified window.

```
enum UIInterfaceOrientation
```

Constants that specify the orientation of the app’s user interface.

```
struct UIInterfaceOrientationMask
```

Constants that specify a view controller’s supported interface orientations.

```
class let invalidInterfaceOrientationException: NSExceptionName
```

An exception that’s thrown if a view controller or the app returns an invalid set of supported interface orientations.

Providing a window for storyboarding

```
var window: UIWindow?
```

The window to use when presenting a storyboard.

Providing the main entry point

```
static func main()
```

Provides the top-level entry point for the app.

Deprecated

~~`func applicationDidFinishLaunching(UIApplication)`~~

Tells the delegate when the app has finished launching.

Deprecated

Deprecated symbols

Symbols that are no longer supported.

Relationships

Inherits From

`NSObjectProtocol`

See Also

Life cycle

Managing your app’s life cycle

Respond to system notifications when your app is in the foreground or background, and handle other significant system-related events.

Responding to the launch of your app

Initialize your app’s data structures, prepare your app to run, and respond to any launch-time requests from the system.

`class UIApplication`

The centralized point of control and coordination for apps running in iOS.

Scenes

Manage multiple instances of your app’s UI simultaneously, and direct resources to the appropriate instance of your UI.