

プッシュ通知処理はSceneDelegateに移行する必要があるか？

Apple公式ドキュメントにおける位置付け

- `UIApplicationDelegate` の概要には、App Delegateはアプリの共有挙動を管理し、**アプリの起動時に必要なサービスを登録する責任があると**記されています。例として「Apple Push Notification service などのサービスへの登録」が挙げられており ¹、プッシュ通知に関する処理が依然としてApp Delegateの役割であることがわかります。
- `UNUserNotificationCenterDelegate` の概要では、通知の選択アクションを処理するために実装したデリゲートオブジェクトは、**アプリが起動を完了する前に共有の `UNUserNotificationCenter` に登録する必要があると明示されています**。具体的には、iOSアプリの場合は `application(_:willFinishLaunchingWithOptions:)` または `application(_:didFinishLaunchingWithOptions:)` メソッド内でデリゲートを設定するよう説明されています ²。Scene Delegateのメソッドは例示されておらず、App Delegateで設定することが推奨されています。

UIKitにおけるScene対応とプッシュ通知の関係

- iOS 26以降では、Sceneベースのライフサイクルが必須となり、`UISceneDelegate` を導入しなければアプリが起動しないことがAppleのテクニカルノートTN3187などで示されています。しかし、この変更は**アプリの起動とUI管理の仕組み**に関するものであり、APNs (Apple Push Notification service) 自体のフローは変わりません。
- Courier社がiOS 26のScene強制化についてまとめた記事では、「**プッシュ通知の基本は変わらない。デバイスのトークン取得やAPNsからのコールバックは引き続き `AppDelegate` に実装する**」と明記しています ³ ⁴。また、Scene対応によってアプリが正しく起動しないとプッシュ通知の登録が行われないため、Scene Delegateを追加する必要はあるが、APNsコールバックの場所は変わらないとしています ⁴。
- Scene Delegateを採用しているアプリでも、プッシュ通知に反応して特定の画面へ遷移したい場合は、`AppDelegate` で受け取った通知コールバックから `SceneDelegate` の `window` やルートビューコントローラを参照してUIを更新するという形になります。iOS 13以降を対象としたチュートリアルでは「**通知タップ時に呼び出されるメソッドは依然としてApp Delegateに存在する**」と説明し、UIの変更には `UIApplication.shared.connectedScenes.first?.delegate` でScene Delegateのwindowにアクセスする方法を紹介しています ⁵。
- Swift Rivalsの移行ガイドでは、`application(_:didReceiveRemoteNotification:fetchCompletionHandler:)` に対するScene Delegateの直接の代替は存在せず、**プッシュ通知処理はアプリ全体の責務としてApp Delegateに実装すると述べています** ⁶。Scene固有の通知を処理したい場合は、Scene Delegate内でUI更新を行うことは可能ですが、通知自体の受信はApp Delegateで行います。

まとめと推奨

- Scene Delegateへの移行はUIとライフサイクル管理に関するものであり、プッシュ通知の受信やデバイス登録の処理をScene Delegateに移動する必要はありません。
- Apple公式ドキュメントでは、`UNUserNotificationCenter`のデリゲート設定やApple Push Notification serviceへの登録をApp Delegateの起動時メソッド内で行うことを推奨しています ¹ ²。
- iOS 26以降も、プッシュ通知のコールバックメソッド（例えば `application(_:didRegisterForRemoteNotificationsWithDeviceToken:)` や

`application(_:didReceiveRemoteNotification:fetchCompletionHandler:)` は `UIApplicationDelegate` の一部であり、Scene Delegateに対応するメソッドはありません⁶。

- Sceneベースライフサイクルを採用したプロジェクトでは、App Delegateで受け取った通知を使ってUIを更新する際に、`UIApplication.shared.connectedScenes` から目的のScene Delegateインスタンスを取得してその `window` やルートビューコントローラにアクセスする方法が推奨されています⁵。
- したがって、プッシュ通知処理をApp DelegateからScene Delegateへ完全に移行する必要はなく、App Delegateに残したままScene Delegateと連携する形で対応するのが適切です。

¹ UIApplicationDelegate | Apple Developer Documentation

<https://developer.apple.com/documentation/uikit/uiapplicationdelegate>

² UNUserNotificationCenterDelegate | Apple Developer Documentation

<https://developer.apple.com/documentation/usernotifications/unusernotificationcenterdelegate>

³ ⁴ iOS 26: UISceneDelegate Soon Required for UIKit. What developers need to know.

<https://www.courier.com/blog/ios-26-push-notification-changes-uiscene-requirement-ios-27>

⁵ Open app in specific view when push notification is tapped (iOS 13+)

<https://fluffy.es/open-app-in-specific-view-when-push-notification-is-tapped-ios-13/>

⁶ Migrating AppDelegate to SceneDelegate | Swift Rivals

<https://swiftrivals.com/blog/migrating-appdelegate-to-scenedelegate>