About Local and Remote Notifications

Important: This documentation contains preliminary information about an API or technology in development. This information is subject to change, and software implemented according to this documentation should be tested with final operating system software.

Local notifications and remote notifications are the two types of so-called *user* notifications, as distinguished from broadcast notifications (managed by the `NSNotificationCenter` class) and key-value observing notifications. User notifications enable an app that isn't running in the foreground to let its users know it has information for them. The information could be a message, an impending calendar event, or new data on a remote server, for example. When presented by the operating system, user notifications, whether local or remote in origin, look and sound the same. They can display an alert message or they can badge the app icon. They can also play a sound when the alert or badge number is shown.

Upon receiving a user notification, the user can tap it to launch the associated app and see the details. They can also choose to ignore the notification, in which case the app is not activated.

At a Glance

Local and remote notifications appear the same to a user, but they serve different use cases and you configure and manage them differently.

Local and Remote Notifications Contrasted

Many apps operate in a time-based or interconnected environment where events of interest to users occur when the app is not in the foreground. Local and remote notifications allow these apps to notify their users when these events occur.

Local and remote notifications serve different design needs, as follows:

- A local notification is scheduled and sent by the app itself, without necessary involvement of the Internet.
- A remote notification, also called a push notification, arrives from outside the device. It originates on a remote server that you manage—the app's *provider*—and is pushed to your app on a user's device via the Apple Push Notification service (APNs).

Relevant Chapter: Local and Remote Notifications in Depth

Registering, Scheduling, and Handling User Notifications

For the system deliver a local notification at a later time, an app registers notification types (in iOS 8 and later), creates a local notification object (using either `UILocalNotification` or `NSUserNotification`), assigns it a delivery date and time, specifies presentation details, and schedules it for delivery. To receive remote notifications, an app must register notification types, then pass to its provider an app-specific device token it gets from the operating system.

When the operating system delivers a local notification or remote notification and the target app is not running in the foreground, it can present the notification to the user through an alert, icon badge number, or sound. If there is a notification alert and the user taps or clicks an action button (or moves the action slider), the app launches and calls a method to pass in the local-notification object or remote-notification payload. If the app is running in the foreground when the notification is delivered, the app delegate receives a local or remote notification.

In iOS 8 and later, user notifications can include custom actions. Also, location-based local notifications can be sent whenever the user arrives at a particular geographic location.

Relevant Chapter: Registering, Scheduling, and Handling User Notifications

Apple Push Notification Service

Apple Push Notification service (APNs) propagates remote notifications to devices having apps registered to receive those notifications. Each app on a device establishes an accredited and encrypted IP connection with the service and receives notifications over this persistent connection. Providers connect with APNs through a persistent and secure channel while monitoring incoming data intended for their client apps. When new data for an app arrives, the provider prepares and sends a notification through the channel to APNs, which pushes the notification to the target device.

The APNs Provider API is asynchronous and, starting in December 2015, uses HTTP/2 to send remote notification requests from your provider server to APNs. The provider composes each outgoing notification and sends it over this channel to APNs.

Related Chapters: Apple Push Notification Service, APNs Provider API

Security Credentials for Remote Notifications

To develop and deploy the remote notification provider server for your app, you must get one or more SSL certificates from Member Center. Starting in December 2015, the HTTP/2-based provider API lets you obtain a single certificate to use for both your development and production environments. In addition, the single certificate can be used to send notifications to not only the primary app (as identified by bundle ID) but also to associated Apple Watch complications and backgrounded VoIP services.

Related Chapter: Provisioning and Development

Prerequisites

App Programming Guide for iOS describes the high level patterns for writing iOS apps.

For local notifications and the client-side implementation of remote notifications, familiarity with app development for iOS is assumed. For the provider side of the implementation, knowledge of TLS/SSL and streaming sockets is helpful.

See Also

The following documents provide background information:

- *App Distribution Quick Start* teaches how to create a team provisioning profile in Xcode before you enable APNs.
- Configuring Push Notifications in *App Distribution Guide* explains the steps for obtaining APNs certificates from Member Center.
- *Entitlement Key Reference* documents the specific entitlements needed for an app to receive

remote notifications.

You might find these additional sources of information useful for understanding and implementing local and remote notifications:

- Notification Essentials in *App Programming Guide for watchOS* explains how Apple Watch supports user notifications.
- The reference documentation for `UILocalNotification`, `UIApplication`, and `UIApplicationDelegate` describe the local- and remote-notification API for client apps in iOS.
- The reference documentation for `NSApplication` and `NSApplicationDelegate Protocol` describe the remote-notification API for client apps in OS X.
- *Security Overview* describes the security technologies and techniques used for the iOS and OS X systems.
- RFC 5246 is the standard for the TLS protocol.

  Secure communication between data providers and Apple Push Notification service requires knowledge of Transport Layer Security (TLS) or its predecessor, Secure Sockets Layer (SSL). Refer to one of the many online or printed descriptions of these cryptographic protocols for further information.

For information on how to send push notifications to your website visitors using OS X, read Configuring Safari Push Notifications in *Notification Programming Guide for Websites*.

For help with issues you encounter with sending or receiving remote (push) notifications, read *Technical Note TN2265*, Troubleshooting Push Notifications.