

[Background Tasks](#) / Choosing Background Strategies for Your App

Article

Choosing Background Strategies for Your App

Select the best method of scheduling background runtime for your app.

Overview

If your app needs computing resources to complete tasks when it's not running in the foreground, you can select from many strategies to obtain background runtime. Selecting the right strategies for your app depends on how it functions in the background.

Some apps perform work for a short time while in the foreground and must continue uninterrupted if they go to the background. Other apps defer that work to perform in the background at a later time or even at night while the device charges. Some apps need background processing time at varied and unpredictable times, such as when an external event or message arrives.

Apps involved in health research studies can obtain background runtime to process data essential for the study. Apps can also request to launch in the background for studies in which the user participates.

Select one or more methods for your app based on how you schedule activity in the background.

Continue Foreground Work in the Background

The system may place apps in the background at any time. If your app performs critical work that must continue while it runs in the background, use `beginBackgroundTask(withName:expirationHandler:)` to alert the system. Consider this approach if your app needs to finish sending a message or complete saving a file.

The system grants your app a limited amount of time to perform its work once it enters the background. Don't exceed this time, and use the expiration handler to cover the case where the time has depleted to cancel or defer the work.

Once your work completes, call `endBackgroundTask(_ :)` before the time limit expires so that your app suspends properly. The system terminates your app if you fail to call this method.

If the task is one that takes some time, such as downloading or uploading files, use `URLSession` (Swift) or `NSURLSession` (Objective-C) . See [Downloading files in the background](#) for more information.

Defer Intensive Work

To preserve battery life and performance, you can schedule background tasks for periods of low activity, such as overnight when the device charges. Use this approach when your app manages heavy workloads, such as training machine learning models or performing database maintenance.

Schedule these types of background tasks using `BGProcessingTask`, and the system decides the best time to launch your background task.

Apps involved in health research studies can have time-consuming tasks essential for the study and might need to complete processing the background. Schedule these types of background tasks using `BGHealthResearchTask`.

Update Your App's Content

Your app may require short bursts of background time to perform content refresh or other work; for example, your app may fetch content from the server periodically, or regularly update its internal state. In this situation, use `BGAppRefreshTask` by requesting `BGAppRefreshTaskRequest`.

Your app can use `BGHealthResearchTaskRequest` to launch in the background and process data for a health research study in which the user participates.

The system decides the best time to launch your background task, and provides your app up to 30 seconds of background runtime. Complete your work within this time period and call `setTaskCompleted(success:)`, or the system terminates your app. See [Background Tasks](#) for more information.

Wake Your App with a Background Push

Background pushes silently wake your app in the background. They don't display an alert, play a sound, or badge your app's icon. If your app obtains content from a server infrequently or at irregular intervals, use background pushes to notify your app when new content becomes available. A messaging app with a muted conversation might use a background push solution, and so might an email app that processes incoming mail without alerting the user.

When sending a background push, set `content-available:` to 1 without alert, sound, or badge. The system decides when to launch the app to download the content. To ensure your app launches, set `apns-priority` to 5, and `apns-push-type` to background.

Once the system delivers the remote notification with `application(_:didReceiveRemoteNotification:fetchCompletionHandler:)`, your app has up to 30 seconds to complete its work. Once your app performs the work, call the passed completion handler as soon as possible to conserve power. If you send background pushes more frequently than three times per hour, the system imposes rate limitations. See [Pushing background updates to your App](#) for more information.

Request Background Time and Notify the User

If your app needs to perform a task in the background and show a notification to the user, use a Notification Service Extension. For example, an email app might need to notify a user after downloading a new email. Subclass `UNNotificationServiceExtension` and bundle the system extension with your app. Upon receiving a push notification, your service extension wakes up and obtains background runtime through `didReceive(_:withContentHandler:)`.

When your extension completes its work, it must call the content handler with the content you want to deliver to the user. Your extension has a limited amount of time to modify the content and execute the `contentHandler` block.

See Also

Background tasks

Using background tasks to update your app

Configure your app to perform tasks in the background to make efficient use of processing time and power.

Refreshing and Maintaining Your App Using Background Tasks

Use scheduled background tasks for refreshing your app content and for performing maintenance.

`class BGProcessingTask`

A time-consuming processing task that runs while the app is in the background.

`class BGAppRefreshTask`

An object representing a short task typically used to refresh content that's run while the app is in the background.

```
class BGHealthResearchTask
```

A time-consuming, necessary processing task that runs while the app is in the background to prepare data essential to a health research study.