

iOS Background Execution Limits



This thread has been locked by a moderator.



5.1k

I regularly see questions, both here on DevForums and in my Day Job™ at DTS, that are caused by a fundamental misunderstanding of how background execution works on iOS. These come in many different variants, for example:

- How do I keep my app running continuously in the background?
- If I schedule a timer, how do I get it to fire when the screen is locked?
- How do I run code in the background every 15 minutes?
- How do I set up a network server that runs in the background?
- How can my app provide an IPC service to another one of my apps while it's in the background?

The short answer to all of these is **You can't**. iOS puts strict limits on background execution. Its default behaviour is to suspend your app shortly after the user has moved it to the background; this suspension prevents the process from running any code.

There's no general-purpose mechanism for:

- Running code continuously in the background
- Running code at some specific time in the background
- Running code periodically at a guaranteed interval
- Resuming in the background in response to a network or IPC request

However, iOS does provide a *wide* range of special-purpose mechanisms for accomplishing specific user goals. For example:

- If you're building a music player, use the `audio background mode` to continue playing after the user has moved your app to the background.
- If you're building a timer app, use a `local notification` to notify the user when your timer has expired.
- If you're building a video player app, use AVFoundation's `download support`.

Keep in mind that the above is just a short list of examples. There are many other special-purpose background execution mechanisms, so you should search the documentation for something appropriate to your needs.

IMPORTANT Each of these mechanisms fulfils a specific purpose. Do not attempt to use them for some other purpose. Before using a background API, read clause 2.5.4 of the [App Store Review Guidelines](#).

Additionally, iOS provides some general-purpose mechanisms for background execution:

- To resume your app in the background in response to an event on your server, use a background notification (aka a 'silent' push). For more information, see [Pushing Background Updates to Your App](#).
- To request a small amount of background execution time to refresh your UI, use `BGAppRefreshTaskRequest`.
- To request extended background execution time, typically delivered overnight when the user is asleep, use `BGProcessingTaskRequest`.
- To prevent your app from being suspended for a short period of time so that you can complete some user task, use a `UIApplication` background task. For more information on this, see [UIApplication Background Task Notes](#).
- To download or upload a large HTTP resource, use an `NSURLSession` `background session`.

All of these mechanisms prevent you from abusing them to run arbitrary code in the background. As an example, consider the `NSURLSession` `resume rate limiter`.

For more information about these limitations, and background execution in general, I *strongly* recommend that you watch WWDC 2020 Session 10063 [Background execution demystified](#). It's an excellent resource.

Specifically, this talk addresses a common misconception about the app refresh mechanism (`BGAppRefreshTaskRequest` and the older background fetch API). Folks assume that app refresh will provide regular background execution time. That's not the case. The system applies a range of heuristics to decide which apps get app refresh time and when. This is a complex issue, one that I'm not going to try to summarise here, but the take-home message is that, if you expect that the app refresh mechanism will grant you background execution time, say, every 15 minutes, you'll be disappointed. In fact, there are common scenarios where it won't grant you *any* background execution time at all! Watch the talk for the details.

Finally, if you have questions about background execution that aren't covered by the resources listed here, please open a new thread on DevForums with the details. Tag it appropriately for the technology you're using; if nothing specific springs to mind, use *Background Tasks*. Also, make sure to include details about the specific problem you're trying to solve because, when it comes to background execution, the devil really is in the details.

Share and Enjoy

Quinn "The Eskimo!" @ Developer Technical Support @ Apple
`let myEmail = "eskimo" + "1" + "@" + "apple.com"`

Change history:

- **2023-05-11** Added a paragraph that explains a common misconception about the app refresh mechanism. Made other minor editorial changes.
- **2021-08-12** Added more entries to the common questions list, this time related to networking and IPC. Made minor editorial changes.
- **2021-07-26** Extended the statement about what's not possible to include "running code periodically at a guaranteed interval".
- **2021-07-22** First posted.

Background Tasks

Reply

Posted 1 year ago by eskimo

Add a Comment

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the [Apple Developer Forums Participation Agreement](#).

Apple > Developer > Forums

Platforms

iOS
iPadOS
macOS
tvOS
watchOS

Tools
Swift
SwiftUI
SF Symbols
Swift Playgrounds
TestFlight
Xcode
Xcode Cloud

Topics & Technologies

Accessibility
Accessories
App Extensions
App Store
Audio & Video
Augmented Reality
Business
Design
Distribution
Education
Fonts
Games
Health & Fitness
In-App Purchase
Localization
Maps & Location
Machine Learning
Security
Safari & Web

Resources

Documentation
Curriculum
Downloads
Forums
Videos

Support
Support Articles
Contact Us
Bug Reporting
System Status

Account
Apple Developer
App Store Connect
Certificates, IDs, & Profiles
Feedback Assistant

Programs

Apple Developer Program
Apple Developer Enterprise Program
App Store Small Business Program
MFi Program
News Partner Program
Video Partner Program
Security Bounty Program
Security Research Device Program

Events
App Accelerators
App Store Awards
Apple Design Awards
Apple Developer Academies
Entrepreneur Camp
Tech Talks
WWDC