

Testing Background Session Code

This thread has been locked by a moderator.



26k

Debugging code that runs in the background on iOS is tricky because the act of running your app in the debugger affects how it behaves. This is especially true with `URLSession` background session code. I regularly see folks confused about how to test and debug such code, and this is my attempt to help with that.

If you have questions or comments about this, start a new thread and tag it with *Foundation* and *CFNetwork* so that I see it.

Share and Enjoy

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

Testing Background Session Code

Testing an app that uses a `URLSession` background session is tricky because of the differences between the test environment and the production environment. This post explains those differences and describes how you can effectively test your background session code.

Confusing Behaviours

When testing background session code you might encounter four common sources of confusion:

- When you run your app from Xcode, Xcode installs the app in a new container, meaning that the path to your app changes. Background sessions *should* deal with this path change correctly, but there have been times in the past where that’s not been the case.
Note If you encounter a problem with a background session mishandling a container path change, please report that as a [bug](#).
- Xcode’s debugger prevents the system from suspending your app. If you run your app from Xcode, or you attach to the process after launch, and then move your app into the background, your app will continue executing in situations where the system would otherwise have suspended it.
- Simulator may not accurately simulate app suspend and resume.
Note This is supposed to work but there have been problems with it in the past (r. 16532261). If you see this problem on modern versions of Xcode, please report it as a [bug](#).
- When you remove an app from the multitasking UI (aka **force quit** it), the system interprets that as a strong indication that the app should do no more work in the background, and that includes `URLSession` background tasks.

Testing Recommendations

Given the above, my recommendation is that you test your app as follows:

- Test on a real device, not in Simulator.
- Run your app from the Home screen rather than running it from Xcode.
- Don’t use force quit to test the ‘relaunch in the background case’.

Doing this avoids all of the issues described above, resulting in a run-time environment that’s much closer to what your users will see.

Debugging Background Session Code

If you encounter problems that you need to debug, you have two options. The first is to use logging. Your app needs good logging support anyway; without that, it’ll be impossible to debug problems that only show up in the field. Once you’ve taken the time to create this logging, use it to debug problems during development.

I recommend that you log to the system log. For more on that, see [Your Friend the System Log](#).

If you have a specific problem that you must investigate with the debugger, run your app from the Home screen and then attach to the running process by choosing Debug > Attach to Process in Xcode. Alternatively, use “Wait for executable to be launched” in the Info tab of the Xcode’s scheme editor.

IMPORTANT As mentioned above, the debugger prevents your app from suspending. If that’s a problem, you can always detach and then reattach later on.

About Force Quit

The behaviour of background sessions after a force quit has changed over time:

- In iOS 7 the background session and all its tasks just ‘disappear’.
- In iOS 8 and later the background session persists but all of the tasks are cancelled.

Note You can use the `NSURLErrorBackgroundTaskCancelledReasonKey` property of the error to find out why a task was cancelled.

Test that your app handles the force quit behaviour of the platforms you support. However, don’t use force quit to test how your app deals with the ‘relaunch in the background case’.

The best way to test the ‘relaunch in the background case’ is to have your app terminate itself by calling `exit`. The system does not interpret this as a force quit, and thus will relaunch your app in the background when your background session needs attention.

There two ways to go about this:

- Simply add a Quit button that calls `exit`.
- Add some setting that causes your app to call `exit` shortly after moving to the background. You’ll need to use a `UIApplication` background task to prevent your app from being suspended before it gets a chance to quit.

IMPORTANT The suggestions above are for testing only. Do not include a Quit button in your final app! This is specifically proscribed by [QA1561](#).

Starting From Scratch

While bringing up a feature that uses background sessions, it’s often useful to start from a clean slate on launch; this prevents tasks from a previous debugging session confusing the current debugging session. You have a number of options here:

- If you’re testing in Simulator, Device > Erase All Content and Settings is an excellent way to wipe the slate completely clean.
- On both Simulator and a real device, you can delete your app. This removes the app, everything in the app’s container, and everything created by the app, including any background sessions.
- You can also take advantage of the `invalidateAndCancel()` [method](#), which will cancel any running tasks and then invalidate the session.

There are a few ways to use `invalidateAndCancel()`:

- During active development it might make sense for your app to call this on launch, guaranteeing that you start with a clean slate every time.
- Alternatively, you could keep track of your app’s install path and call it on launch if the install path has changed.
- You might add a ‘hidden’ UI that invalidates the session. This is helpful when you encounter a problem and want to know whether it’s caused by some problem with the background session’s persistent state.

Revision History

- 2023-07-12** Fixed the formatting. Made significant editorial changes.
- 2018-05-03** Added a discussion of force quit. Also made major editorial changes, restructuring the document to be easy to read.
- 2015-08-18** First posted.

(r. 22323379)

Foundation

CFNetwork

Reply

Posted 7 years 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](#).

<ul style="list-style-type: none"> Forums 			
Platforms	Topics & Technologies	Resources	Programs
iOS	Accessibility	Documentation	Apple Developer Program
iPadOS	Accessories	Curriculum	Apple Developer Enterprise Program
macOS	App Extensions	Downloads	App Store Small Business Program
tvOS	App Store	Forums	MFi Program
watchOS	Audio & Video	Videos	News Partner Program
	Augmented Reality		Video Partner Program
Tools	Business	Support	Security Bounty Program
Swift	Design	Support Articles	Security Research Device Program
SwiftUI	Distribution	Contact Us	
SF Symbols	Education	Bug Reporting	
Swift Playgrounds	Fonts	System Status	
TestFlight	Games		
Xcode	Health & Fitness	Account	
Xcode Cloud	In-App Purchase	Apple Developer	App Accelerators
	Localization	App Store Connect	App Store Awards
	Maps & Location	Certificates, IDs, & Profiles	Apple Design Awards
	Machine Learning	Feedback Assistant	Apple Developer Academies
	Security		Entrepreneur Camp
	Safari & Web		Tech Talks
			WWDC

To view the latest developer news, visit

[News and Updates](#)