

Testing Background Session Code

This thread has been locked. Questions are automatically locked after two months of inactivity, or sooner if deemed necessary by a moderator.



21k

Testing an app that uses an

NSURLSession

background session is tricky because there are 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 will 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. So, 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.
- The iOS 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 terminate an app via the multitasking UI (**force quit**), the system interprets that as a strong indication that the app should do no more work in the background, and that includes

NSURLSession

background tasks.

Testing Recommendations

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

- Test on a real device, not the simulator
- Run your app from the Home screen rather than running it from Xcode
- Do not 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:

- Use logging — It's important that your app have good logging support anyway, because otherwise it'll be impossible to debug problems that only show up in the field. Once you've taken the time to create this logging, you can use it to debug problems during development.
- Attach — If you have a specific problem that you must investigate with the debugger, you can run your app from the Home screen and then attach to the running process (via Xcode's Debug > Attach to Process command) or use "Wait for executable to be launched" in the info tab of the 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

NSErrorBackgroundTaskCancelledReasonKey

property of the error to find out why a task was cancelled.

You should test that your app handles the force quit behaviour of the platforms you support. However, you must not 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 you can go about this:

- You can simply add a Quit button that calls

exit

- You can 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 in that case).

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 it's often useful to start from a clean slate on launch; this prevents tasks from a previous debugging session from confusing the current debugging session. You have a couple of options here:

- If you're developing on the simulator, iOS Simulator > Reset Content and Settings is an excellent way to wipe the slate completely clean.
- On both the simulator and a real device, you can delete your app. This will not only remove the app and everything in the app's container, but will also remove any background sessions created by the app.
- You can also take advantage of the

-invalidateAndCancel

method, which will cancel any running tasks and then invalidate the session. There's a few ways to use this:

- During active development it might make sense for your app to call this on launch, guaranteeing that you start with a clean slate.
- Alternatively, you could keep track of your app's install path and call it on launch if the install path has changed.
- You could have 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.

Share and Enjoy — Quinn "The Eskimo!" Apple Developer Relations, Developer Technical Support, Core OS/Hardware

let myEmail = "eskimo" + "1" + "@apple.com"

(r. 22323379)

Change History

- 18 Aug 2015 — First posted.
- 3 May 2018 — Added a discussion of force quit. Also made major editorial changes, restructuring the document to be easy to read.

Network

Asked 5 years ago by [eskimo](#)  

Reply to this question

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](#).