

Pre iOS13 - Background Tasks

Mohammad Faani

Overview

- Problem
- Overview of Closures
- How it relates to background task
- API Considerations
- Real demo (with Console)

Sync Closure

```
func processTwoNumbers(_ num1: Int, _ num2: Int,  
    handler: (Int, Int) -> T) {  
    handler(num1, num2)  
}
```

```
processTwoNumbers(10, 22, handler: { num1, num2 in  
    print(num1 + num2)  
})
```

The async closure — block isn't stored.

```
enum Result {  
    case success(Data?)  
    case failure(Error?)  
}  
  
func startFlow(handler: @escaping (Result) -> ()) {  
    let url = URL(string: "https://i.imgur.com/7oZTmIC.jpeg")!  
    let task = URLSession.shared.dataTask(with: url) { data, response, error in  
        if error == nil {  
            handler(.success(data))  
        } else {  
            handler(.failure(error))  
        }  
    }  
    task.resume()  
}
```

The async closure — block is stored.

```
enum Result {  
    case success  
    case error  
}  
  
class FlowManager {  
    var _exitHandler: ((Result) -> ())?  
    func startFlow(exitHandler: @escaping  
    ()(Result) -> ()) {  
        _exitHandler = exitHandler  
    }  
  
    func handleFlowExit(result: Result) {  
        _exitHandler?(result)  
    }  
}
```

```
class Controller {  
    var flowManager = FlowManager()  
    func start() {  
        { [weak self] result in  
            switch result {  
            case .success: self?.showSuccessScreen() // Line A  
            case .error: self?.showFailureScreen // Line B  
            }  
        }  
    }  
    func showSuccessScreen() {  
    }  
    func showFailureScreen() {  
    }  
}
```

Question: When do Lines A, B get called? 🤔🤔🤔

What you should really think of the API

```
UIApplication.shared.tellOSIhaveATask(expirationHandler_whichGetsSt  
ored_andToBeCalledIfNeeded_inAbout30Seconds: {  
    immediatelyStopTask()  
    UIApplication.shared.endBackgroundTask(backgroundTaskID)  
})
```

Begin assertion

Devices

RESML-1723063

IPad-DMPX91NSJ2D1

Reports

Crash Reports

Spin Reports

Log Reports

Diagnostic Reports

Mac Analytics Data

system.log

Console

1,423 messages

PROCESS Background

1,423 messages

Pause Now Activities Clear Reload Info Share

All Messages Errors and Faults Save

Message	Category	Type	Subsystem	Time
Deactivation reason added: 11; deactivation reasons: 4128 -> 6176; animati	Application		com.apple.UIKit	09:48:27.434717-0500
-[UIApplication _beginBackgroundTaskWithName:expirationHandler:]: Will add	BackgroundTask		com.apple.UIKit	09:48:27.434823-0500
_UIBackgroundTaskInfo *_addBackgroundTask(NSString *__strong, void (^__str	BackgroundTask		com.apple.UIKit	09:48:27.435184-0500
_UIBackgroundTaskInfo *_addBackgroundTask(NSString *__strong, void (^__str	BackgroundTask		com.apple.UIKit	09:48:27.435236-0500
Deactivation reason removed: 5; deactivation reasons: 6176 -> 6144; animat	Application		com.apple.UIKit	09:48:27.435482-0500
-[UIApplication _beginBackgroundTaskWithName:expirationHandler:]: Will add	BackgroundTask		com.apple.UIKit	09:48:27.436773-0500
_UIBackgroundTaskInfo *_addBackgroundTask(NSString *__strong, void (^__str	BackgroundTask		com.apple.UIKit	09:48:27.436902-0500
_UIBackgroundTaskInfo *_addBackgroundTask(NSString *__strong, void (^__str	BackgroundTask		com.apple.UIKit	09:48:27.437160-0500
Performing snapshot request 0x2802774e0 (type 1)	Common		com.apple.FrontBoard	09:48:27.441349-0500
Snapshot request 0x2802774e0 complete	Common		com.apple.FrontBoard	09:48:27.448413-0500
Performing snapshot request 0x28024cf30 (type 1)	Common		com.apple.FrontBoard	09:48:27.490326-0500
Snapshot request 0x28024cf30 complete	Common		com.apple.FrontBoard	09:48:27.548583-0500
Performing snapshot request 0x280243c30 (type 1)	Common		com.apple.FrontBoard	09:48:27.558490-0500
Snapshot request 0x280243c30 complete	Common		com.apple.FrontBoard	09:48:27.652482-0500
Performing snapshot request 0x280276610 (type 1)	Common		com.apple.FrontBoard	09:48:27.654814-0500
Snapshot request 0x280276610 complete	Common		com.apple.FrontBoard	09:48:27.723108-0500
-[UIApplication _endBackgroundTask:]: Ending background task with UIBackgr	BackgroundTask		com.apple.UIKit	09:48:27.730160-0500
-[UIApplication _endBackgroundTask:]: Ending task with identifier 3 and de	BackgroundTask		com.apple.UIKit	09:48:27.742082-0500
-[UIApplication _endBackgroundTask:]_block_invoke: Decrementing reference	BackgroundTask		com.apple.UIKit	09:48:27.742232-0500
[0x28272a530] [keyboardFocus] Disabling event deferring records requested:	EventDeferring		com.apple.UIKit	09:48:27.748945-0500
sceneOfRecord: sceneID: sceneID:com.mfaani.Background-default persistentI	Applifecycle		com.apple.UIKit	09:48:27.749074-0500
timer!	test		com.mfaani.Background	09:48:27.753617-0500

Background (UIKitCore) Volatile
Subsystem: com.apple.UIKit Category: BackgroundTask Details
2022-02-23 09:48:27.434823-0500

-[UIApplication _beginBackgroundTaskWithName:expirationHandler:]: Will add backgroundTask with taskName: _UIRemoteKeyboard XPC disconnection, expirationHandler: (null)

If you don't end the task

Received process assertions expiration warning!	general		com.apple.runningboard	09:50:40.875883-0500
Notifying client of imminent expiration of assertion	general		com.apple.runningboard	09:50:40.875993-0500
Expiration notification complete	general		com.apple.runningboard	09:50:40.877928-0500
void _fireBackgroundExpirationHandlers(UIApplication *__strong): Firing ba	BackgroundTask		com.apple.UIKit	09:50:40.878186-0500
void _fireBackgroundExpirationHandlers(UIApplication *__strong)_block_invo	BackgroundTask		com.apple.UIKit	09:50:40.878427-0500
void _fireBackgroundExpirationHandlers(UIApplication *__strong): Calling e	BackgroundTask		com.apple.UIKit	09:50:40.878746-0500
Background task still not ended after expiration handlers were called: <_U	BackgroundTask	●	com.apple.UIKit	09:50:40.878988-0500
timer!	test		com.mfaani.Background	09:50:40.995943-0500
timer!	test		com.mfaani.Background	09:50:41.902734-0500
timer!	test		com.mfaani.Background	09:50:42.901125-0500
timer!	test		com.mfaani.Background	09:50:43.900793-0500
timer!	test		com.mfaani.Background	09:50:44.902508-0500
timer!	test		com.mfaani.Background	09:50:45.902328-0500

Background (UIKitCore)

Subsystem: com.apple.UIKit Category: BackgroundTask [Details](#)

Volatile **ERROR**

2022-02-23 09:50:40.878988-0500

Background task still not ended after expiration handlers were called: <_UIBackgroundTaskInfo: 0x282359c80>: taskID = 1, taskName = Called by Background, from \$s10Background14ViewController11viewDidLoadyyF, creationTime = 57865 (elapsed = 56). This app will likely be terminated by the system. Call UIApplication.endBackgroundTask(_:) to avoid this.

If you end the task

timer!	test	com.mfaani.Background	09:36:47.830174-0500
timer!	test	com.mfaani.Background	09:36:48.804706-0500
Received process assertions expiration warning!	general	com.apple.runningboard	09:36:49.572362-0500
Notifying client of imminent expiration of assertion	general	com.apple.runningboard	09:36:49.572808-0500
Expiration notification complete	general	com.apple.runningboard	09:36:49.575769-0500
void _fireBackgroundExpirationHandlers(UIApplication *__strong): Firing ba	BackgroundTask	com.apple.UIKit	09:36:49.575984-0500
void _fireBackgroundExpirationHandlers(UIApplication *__strong)_block_invo	BackgroundTask	com.apple.UIKit	09:36:49.576412-0500
void _fireBackgroundExpirationHandlers(UIApplication *__strong): Calling e	BackgroundTask	com.apple.UIKit	09:36:49.576805-0500
-[UIApplication _endBackgroundTask:]: Ending background task with UIBackgr	BackgroundTask	com.apple.UIKit	09:36:49.577188-0500
-[UIApplication _endBackgroundTask:]: Ending task with identifier 1 and de	BackgroundTask	com.apple.UIKit	09:36:49.577485-0500
-[UIApplication _endBackgroundTask:]_block_invoke: Decrementing reference	BackgroundTask	com.apple.UIKit	09:36:49.577831-0500
-[UIApplication _endBackgroundTask:]_block_invoke: Will invalidate asserti	BackgroundTask	com.apple.UIKit	09:36:49.578147-0500
timer!	test	com.mfaani.Background	09:36:49.779778-0500
timer!	test	com.mfaani.Background	09:36:50.736864-0500
timer!	test	com.mfaani.Background	09:36:51.734374-0500
timer!	test	com.mfaani.Background	09:36:52.734239-0500

Summary

- Tell the OS you need time
- Once you're done, tell the OS you're done
- If you're about to run out of time, the OS will warn you.
 - You should end your task && tell the OS you're done
 - Otherwise the OS will terminate the app and that's bad...
- The function signature is weird, because you associate the block with the begin function, but that's not really what's happening

References

- <https://mfaani.com/posts/uiapplication-backgroundtasks-through-the-lens-of-closures/>
- <https://developer.apple.com/forums/thread/85066>