

Erledigt

Brauche mal Swift Unterstützung

Beitrag von „Sascha_77“ vom 6. Juni 2019, 16:52

Komme hier bei einer Swift Sache nicht weiter. Ich habe diesen Code hier.

Code

```
1. let process = Process()
2. process.launchPath = "/bin/bash"
3. process.arguments = [path] + args
4. let outputPipe = Pipe()
5. let fileHandler = outputPipe.fileHandleForReading
6. process.standardOutput = outputPipe
7.
8. fileHandler.readabilityHandler = { pipe in
9. let data = pipe.availableData
10. if let line = String(data: data, encoding: .utf8) {
11. DispatchQueue.main.async {
12. self.logger.string += line
13. }
14. } else {
15. print("Error decoding data: \(data.base64EncodedString())")
16. }
17. }
18.
19. process.launch()
20. process.waitUntilExit()
21. fileHandler.readabilityHandler = nil
22. }
```

Alles anzeigen

Nachteil ist, dass er während des Ausführens die GUI der App einfriert aus der das ganze gestartet wird. Nun habe ich dieses Beispiel im Netz gefunden um den Task von der GUI abzukoppeln. Nur irgendwie krieg ichs nicht hin mir das entsprechend umzubauen. Vllt. kann mir da ja jemand weiterhelfen? Mit den 3 var's am Anfang ist klar das die ganz nach oben müssen in meiner AppDelegate.swift aber der Rest? Hier das Beispiel:

Code

```
1. var task:Process!
```

```
2. var out:FileHandle?
3. var outputTimer: Timer?
4.
5. let path = "/bin/bash"
6. let workDir = "/path/to/working/folder"
7. let sourcePath = "source"
8. let destinationPath = "destination"
9.
10. let arguments = ["backup.sh", sourcePath, destinationPath]
11. task = Process.init()
12. task.launchPath = path
13. task.arguments = arguments
14. task.currentDirectoryPath = workDir
15.
16. self.task.launch()
17. DispatchQueue.global().async {
18. // this runs in a worker thread, so the UI remains responsive
19. print ("wating for exit")
20. self.task.waitUntilExit()
21. DispatchQueue.main.async {
22. // do so in the main thread
23. if let timer = self.outputTimer {
24. timer.invalidate()
25. self.outputTimer = nil
26. }
27. }
28. }
```

Alles anzeigen