

実演でわかる！

async/await や actor の
iOS アプリへの取り入れ方

Yuta Koshizawa @koher

Xcode 13.2

- 今週リリースされた
- Swift Concurrency の back deployment
 - iOS 13 以降で `async/await` や `actor` が使えるように¹

¹ https://developer.apple.com/documentation/xcode-release-notes/xcode-13_2-release-notes



Use Swift Concurrency #265

Changes from all commits

File filter

Conversations

Jump to



0 / 30 files viewed



Review changes

53

Shared/Repository/Monsters/MonstersRepository.swift

Viewed

...

@@ -9,7 +9,7 @@

9

10

11

12

13

14

15

9

10

11

12

13

14

15

@@ -18,35 +18,32 @@

18

19

20

21

22

23

24

25

26

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

27

28

29

30

31

35

36

37

38

39

Check warning on line 34 in Shared/Repository/Monsters/MonstersRepository.swift

Codecov

/ codecov/patch

Shared/Repository/Monsters/MonstersRepository.swift#L34

Added line #L34 was not covered by tests

今年の振り返り

- Swift Zoomin' #5: 先取り！ Swift 6 の `async/await` (1月)
- `async/await` や `actor` で iOS アプリ開発がどう変わるか
Before & After の具体例で学ぶ (9月)
- Swift Zoomin' #8: Actor ハンズオン (10月)

async/await や actor を
どうやって iOS アプリに取り入れるか

実際に見るのが一番早い

ライブコーディング

今日扱うアプリ

- API を叩いてユーザーの JSON を取得
 - <https://koherent.org/fake-service/api/user?id=1234>
- JSON をデコードしてユーザー情報を表示
- JSON の中に記載された URL からアイコンをダウンロード
- ユーザーのアイコンも表示

今日のポイント

- Other Swift Flags に `-Xfrontend -warn-concurrency`
- コールバックから `async` への変換には `Continuation`
- 非同期関数を始めるところでは `Task.init`
- SwiftUI & iOS 15 以降では `.task`³

³ <https://developer.apple.com/videos/play/wwdc2021/10019/>

今日のポイント

- ObservableObject には @MainActor を付与
 - ObservableObject 側は async メソッドにし、View 側で Task.init (または .task) ³
- View Controller や View には @MainActor を付与

³ <https://developer.apple.com/videos/play/wwdc2021/10019/>