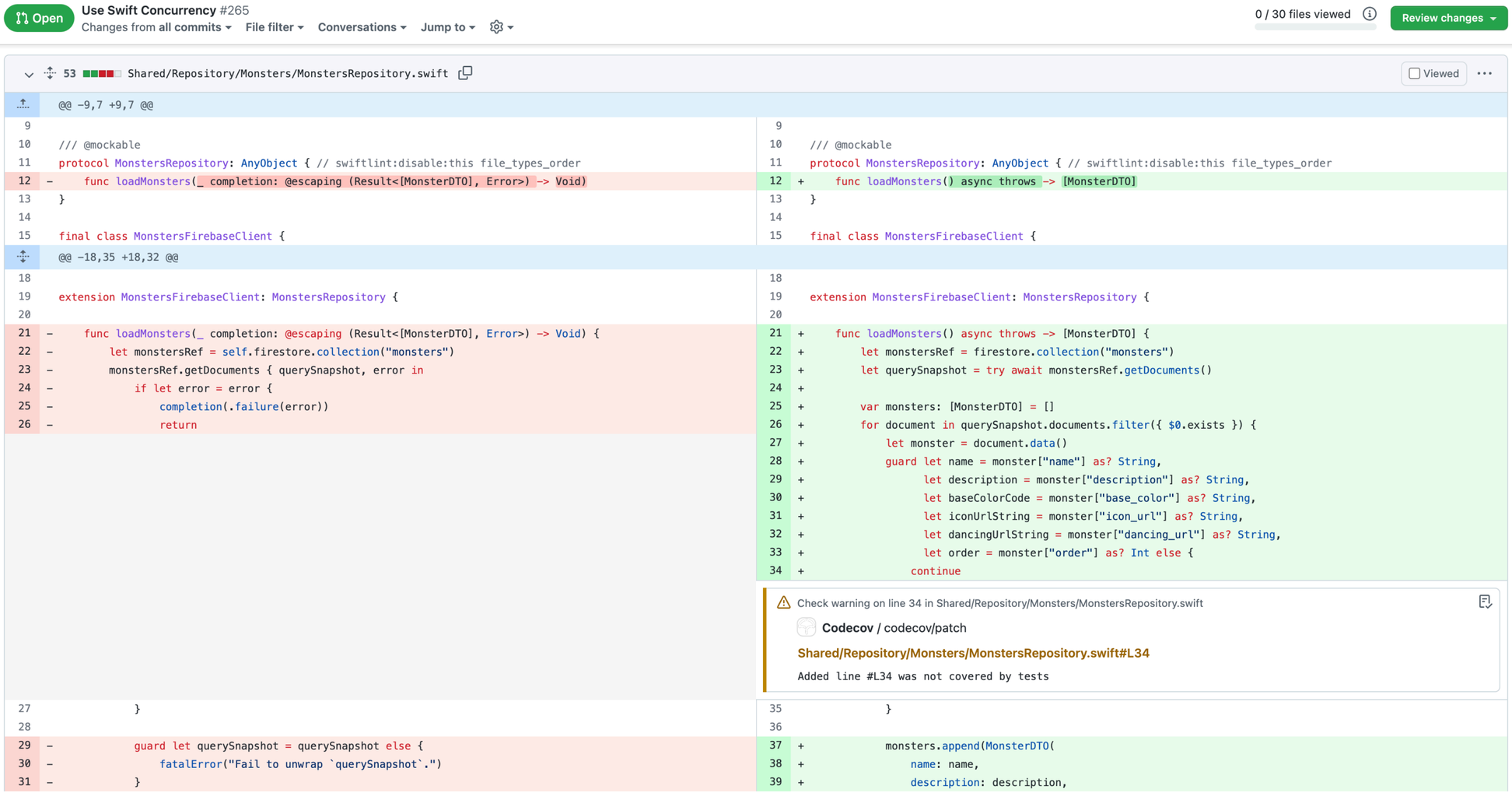
実演でわかる！
`async/await` や `actor` の
iOS アプリへの取り入れ方

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



[2] https://github.com/uhooi/UhooiPicBook/pull/265/files

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

- メインスレッドで実行するときは `MainActor.run`
- Actor 境界を越える型は `Sendable` に適合させる
- `ObservableObject` には `@MainActor` を付与
- View Controller や View には `@MainActor` を付与