

Maximizing Productivity with the iOS Third-Party Ecosystem

Jonas Schlabertz, 30.11.2023

Fastlane

- <https://fastlane.tools/>
- Tool to automate release workflows (Linting, Testing, Building, Signing, Deployment, Screenshots, ...)
- You write a script which defines your different workflows
- Fastlane provides easy-to-use functions for building, signing, altering files or configurations, ...
- Individual workflows can be triggered via CLI

Periphery

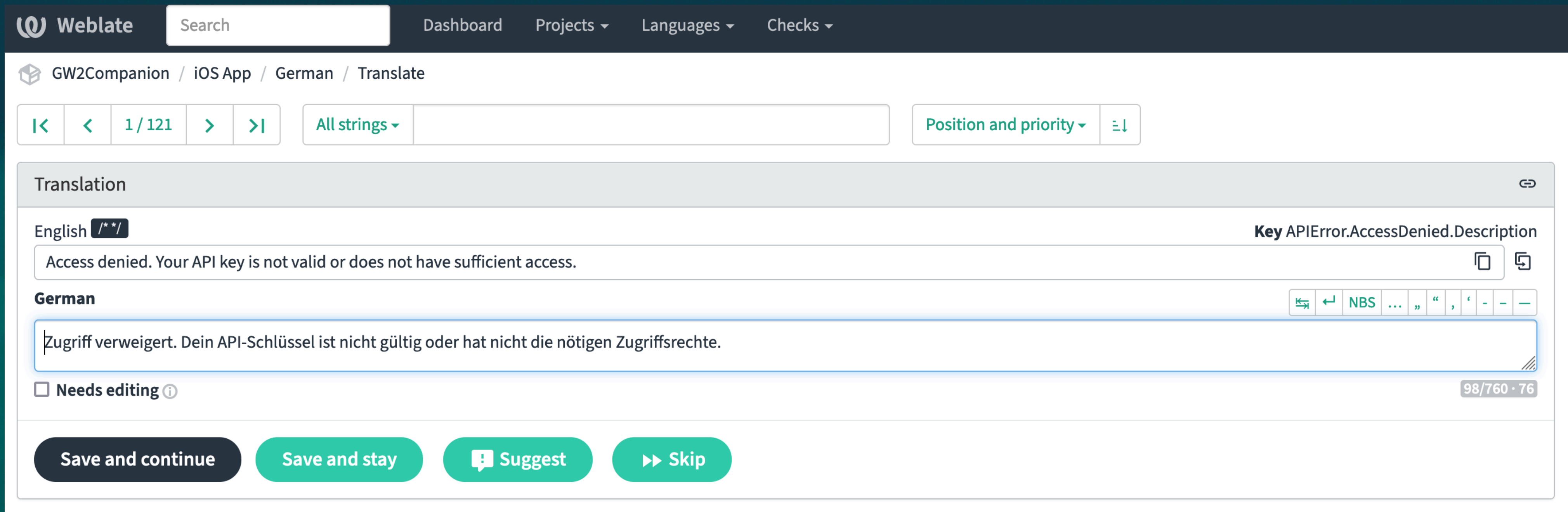
- <https://github.com/peripheryapp/periphery>
- Find unused code in Swift projects
- Analyzes build artifacts to identify unused code
- Xcode Integration: Get a warning for unused code
- CI integration: Can export codeclimate reports for use with e.g. GitLab

SwiftLint

- <https://github.com/realm/SwiftLint>
- Tool to enforce Swift code style, integration into CIs possible
- Code styling rules configurable, new rules can be defined via e.g. regex
- Violations can either throw a warning or an error, thus preventing compilation
- Notable rules:
 - `line_length` (restrict number of characters in a single line)
 - `todo` (`// TODO` comments throw a warning)
 - `whitespace` rules around braces, commas etc.
 - `force_cast` or `force_try`
 - `weak_delegate` (all “delegate” variables need to be weak, avoiding retain cycles)
 - ...



- <https://weblate.org/de/>
- Web Management tool for translated texts (can be self-hosted)
- Exports to (e.g.) Localizable.strings files, wide variety of formats supported
- GitLab/Github integration: Automatic Merge Requests with updated strings files



The screenshot displays the Weblate web interface. At the top, there is a navigation bar with the Weblate logo, a search bar, and links to Dashboard, Projects, Languages, and Checks. Below this, the breadcrumb path is 'GW2Companion / iOS App / German / Translate'. The main area shows a translation for the key 'APIError.AccessDenied.Description'. The English source text is 'Access denied. Your API key is not valid or does not have sufficient access.' The German target text is 'Zugriff verweigert. Dein API-Schlüssel ist nicht gültig oder hat nicht die nötigen Zugriffsrechte.' The interface includes navigation controls (1/121), a filter dropdown (All strings), and a 'Position and priority' dropdown. At the bottom, there are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A status bar at the bottom right shows '98/760 · 76'.

Tuist



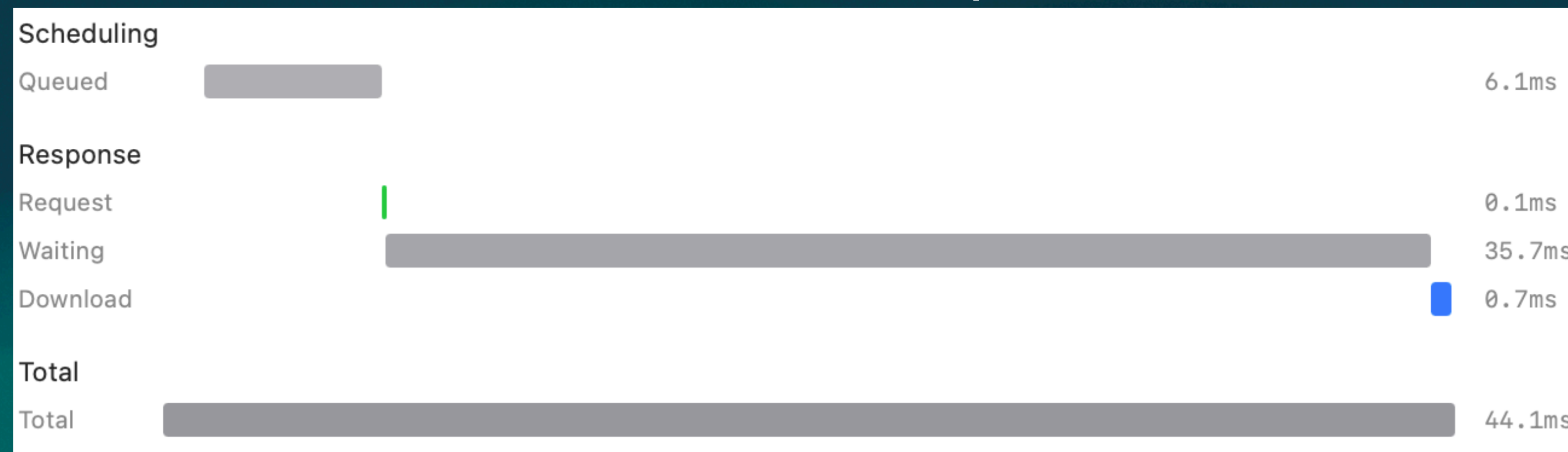
- <https://tuist.io/>
- Idea: Delete your xcworkspace and xcodeproj folders, generate them instead when needed
- Project files defined in a Project.swift file
- Alternative: XcodeGen

```
import ProjectDescription

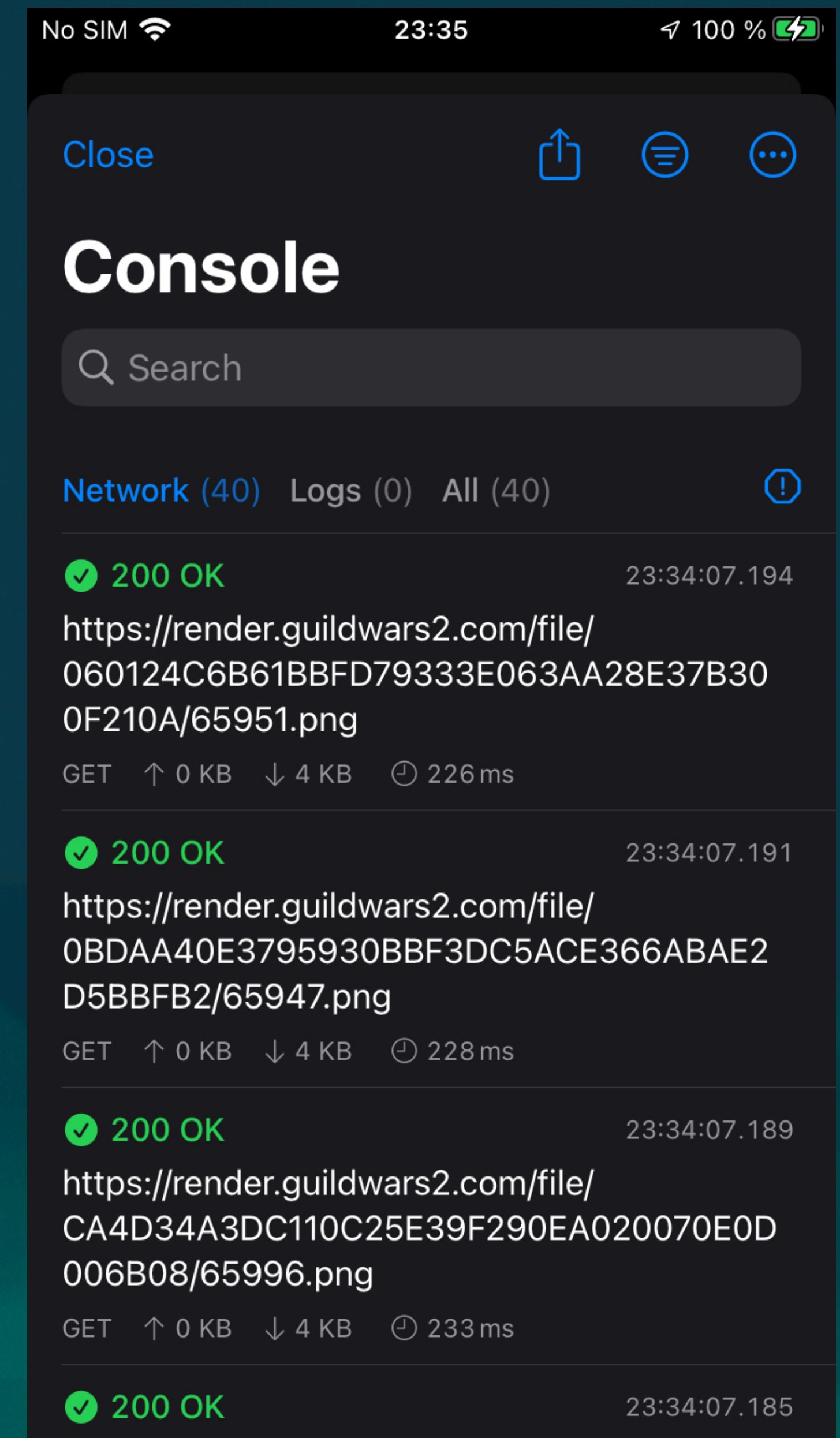
let project = Project(
  name: "DemoApp",
  organizationName: "DemoOrganization",
  targets: [
    Target(
      name: "DemoApp",
      platform: .iOS,
      product: .app,
      bundleId: "com.example.DemoApp",
      infoPlist: "Info.plist",
      sources: ["Sources/**"],
      resources: ["Resources/**"],
      dependencies: []
    )
  ]
)
```


Pulse

- <https://github.com/kean/Pulse>
- Network request logger for Apple platforms
- Requests can be viewed within the app or streamed live to a Mac App
- Provides metrics for each request



- Alternatives: [Charles Proxy](#), [Bagel](#), [mitmproxy](#)



R.swift

- <https://github.com/mac-cain13/R.swift>
 - Access resources (strings, images, XIBs, ...) strongly typed
-
- `UIImageView(image: UIImage(named: "example.image"))`
 - + `UIImageView(image: R.image.exampleImage())`
 - `label.text = NSLocalizedString("general.settings", comment: "")`
 - + `label.text = R.string.localizable.generalSettings()`
-
- R.swift file generated via build phase
 - If an asset is removed but still accessed in code, the build will fail

SwiftGen

- <https://github.com/SwiftGen/SwiftGen>
- Similar to R.swift
- Generates Swift structs to access localized strings and assets

```
Text(L10n.Search.title)
```

```
Image(asset: Asset.Images.exampleImage)
```

- Different asset types split between different structs/files

Kingfisher KINGFISHER

- <https://github.com/onevcats/Kingfisher>
- Library to load images from a server and display them
- Automatically caches images to reduce traffic (heavily customizable)
- Provides image processors and filters to alter displayed images
- UIKit:
`imageView.kf.setImage(with: url)`
- SwiftUI:
`KFImage(url)`
- Alternative: (SwiftUI AsyncImage)

SnapKit

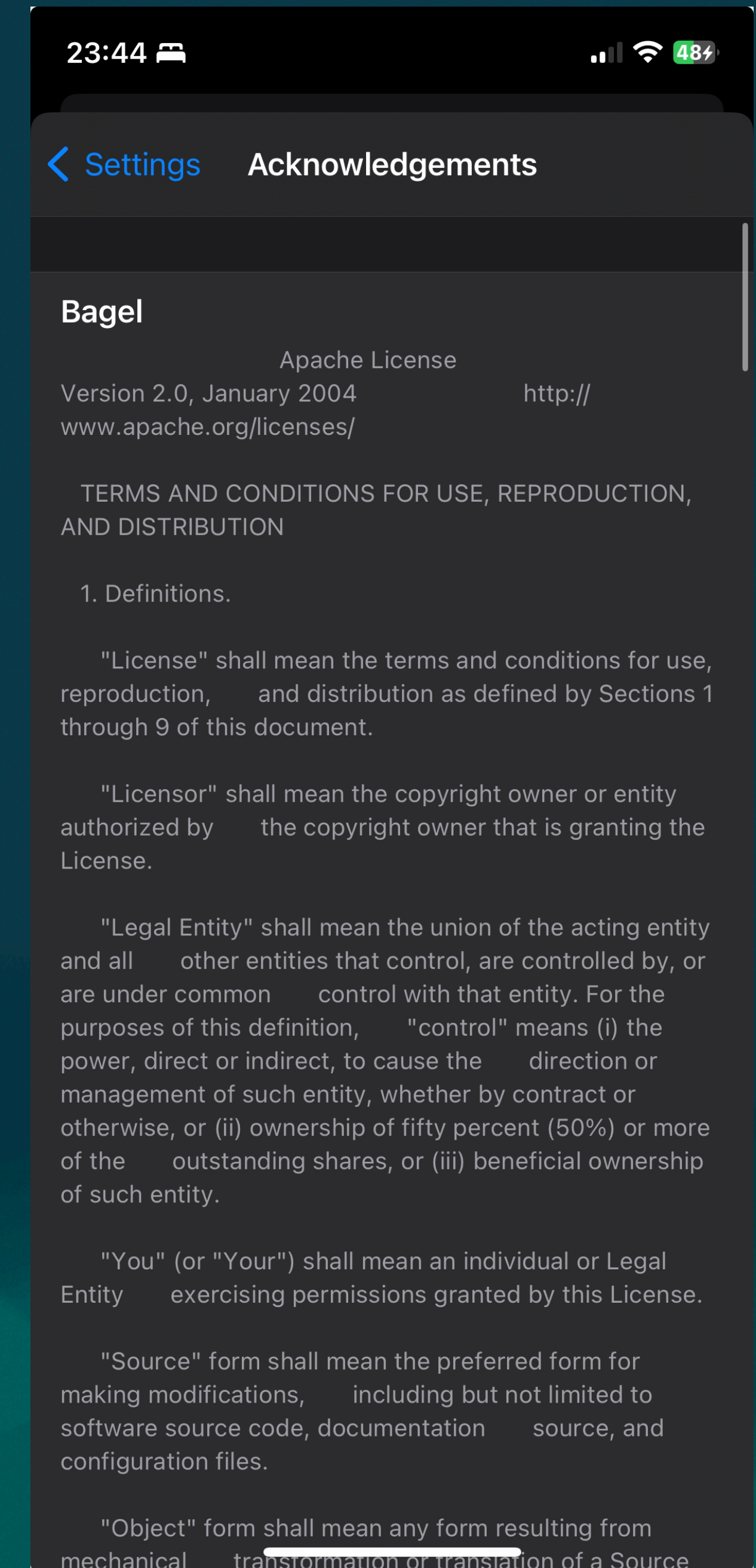
- <https://github.com/SnapKit/SnapKit>
- Wrapping framework around NSLayoutConstraint creation
- Provides a DSL to describe layout constraints

```
nameLabel.snp.makeConstraints { make in
    make.top.equalToSuperview()
    make.leading.equalTo(progressIcon.snp.trailing).offset(8)
    make.trailing.lessThanOrEqualToSuperview().inset(8)
}
```

```
descriptionLabel.snp.makeConstraints { make in
    make.top.equalTo(nameLabel.snp.bottom).offset(8)
    make.leading.equalTo(iconView.snp.trailing).offset(8)
    make.trailing.lessThanOrEqualToSuperview().inset(8)
}
```


LicenseGenerator iOS

- <https://github.com/carloe/LicenseGenerator-iOS>
- Licenses of open source dependencies typically need to be included in an App
- LicensesViewController generates a ViewController with all licenses from all dependencies
- Python script generates ViewController during build phase
- Alternative: LicensePlist



Any Questions?

Jonas Schlabertz
jonas@schlabertz.de
CocoaHeads Aachen, 30.11.2023