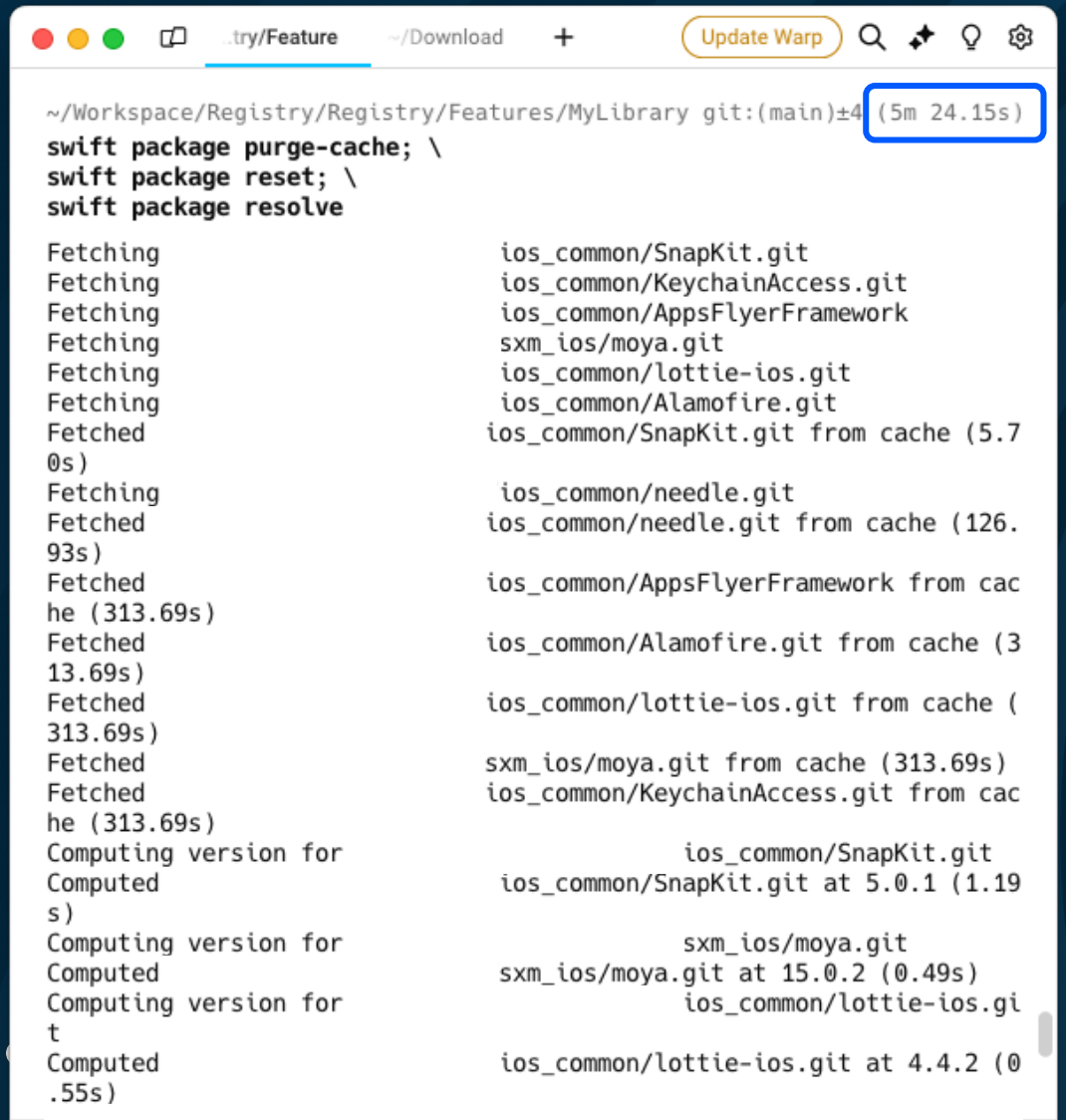
OZON tech

Нервное смузи из Swift Package Registry

Евгений Рыжов
Senior iOS Developer, Ozon

Кому полезно

И каким образом

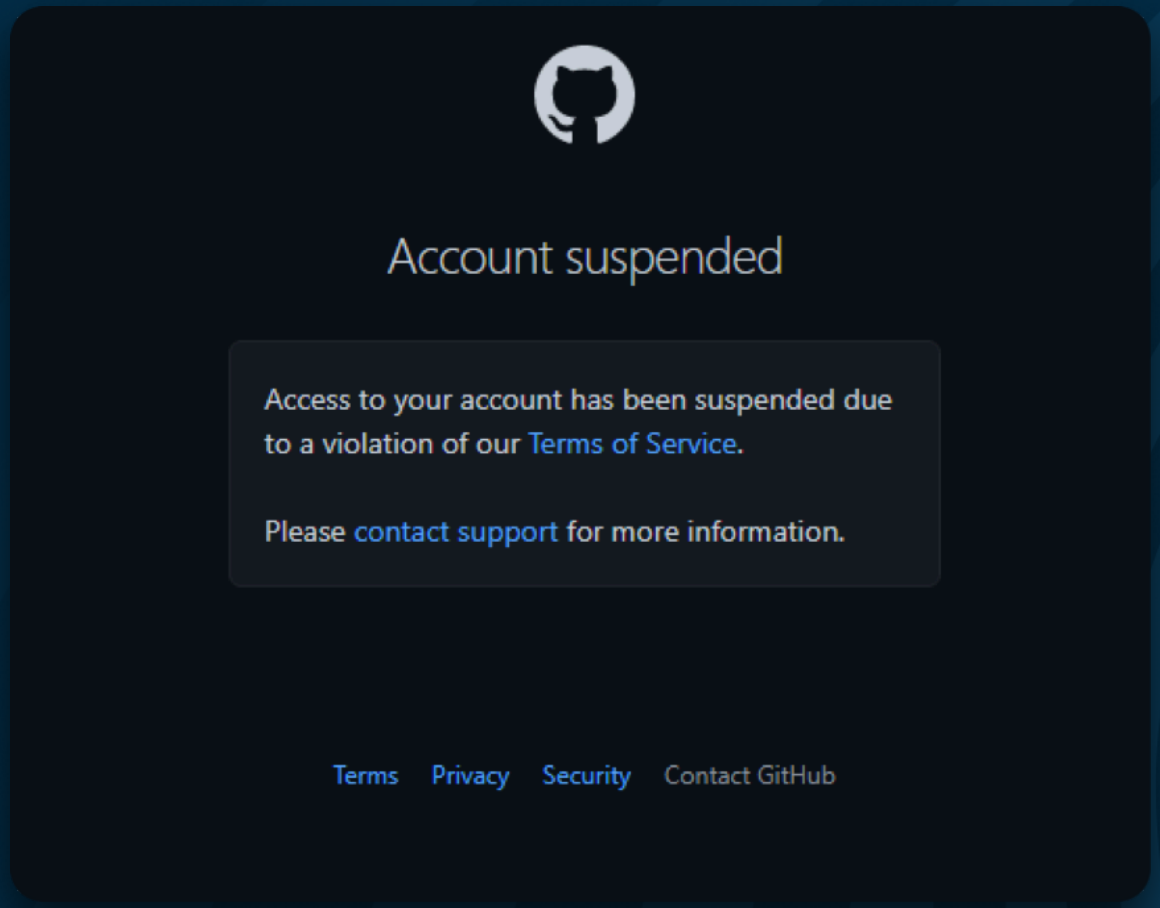


Кому полезно

И каким образом

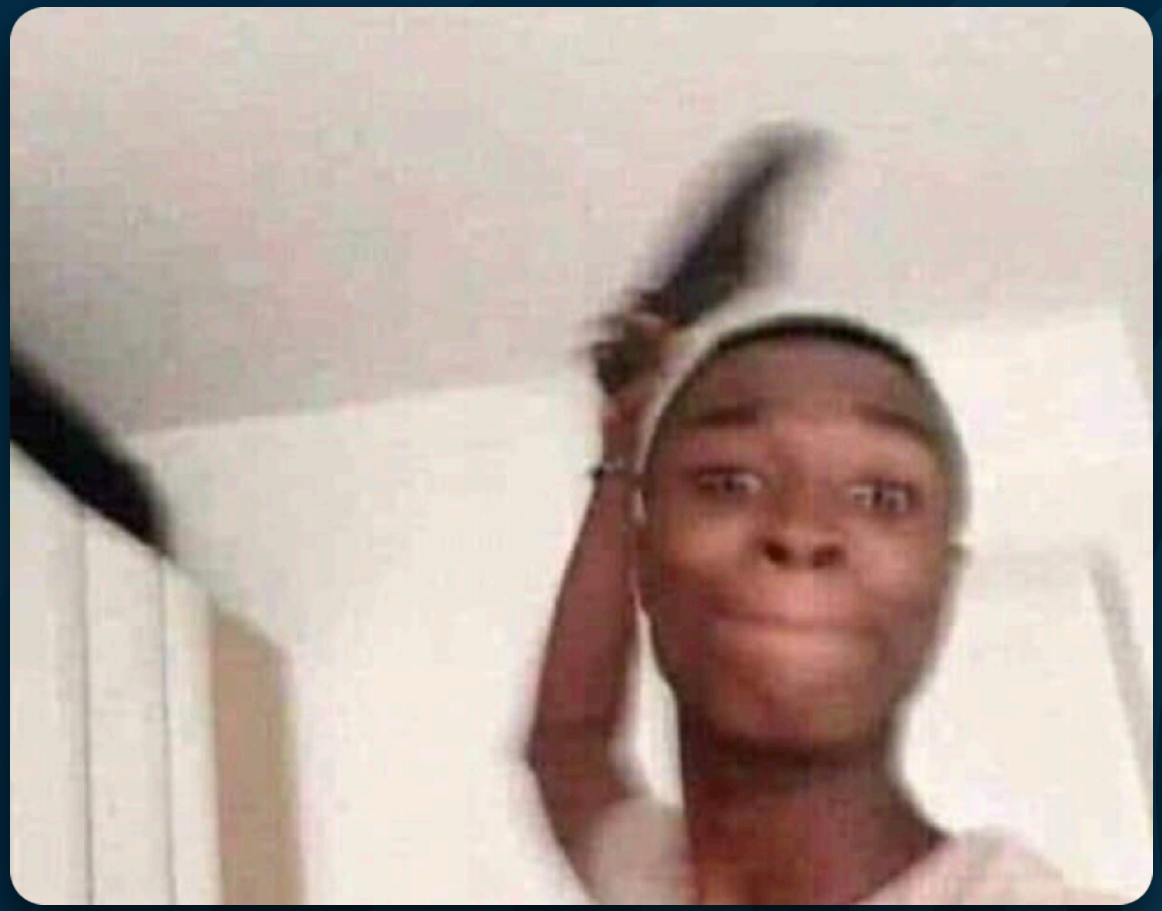
Кому полезно

И каким образом



Кому полезно

И каким образом



Обо мне

Живу в платформе три года без перерывов

В основном работаю с билд системой и менеджерами зависимостей

Не люблю менеджеры зависимостей

AGENDA

- **Swift Package и Registry**
- Swift Registry Service и как с ней работать
- В чём бенефиты и где нюансы
- Заключение. Как оно?

01

Swift Package
и Registry

Что за Swift Package?

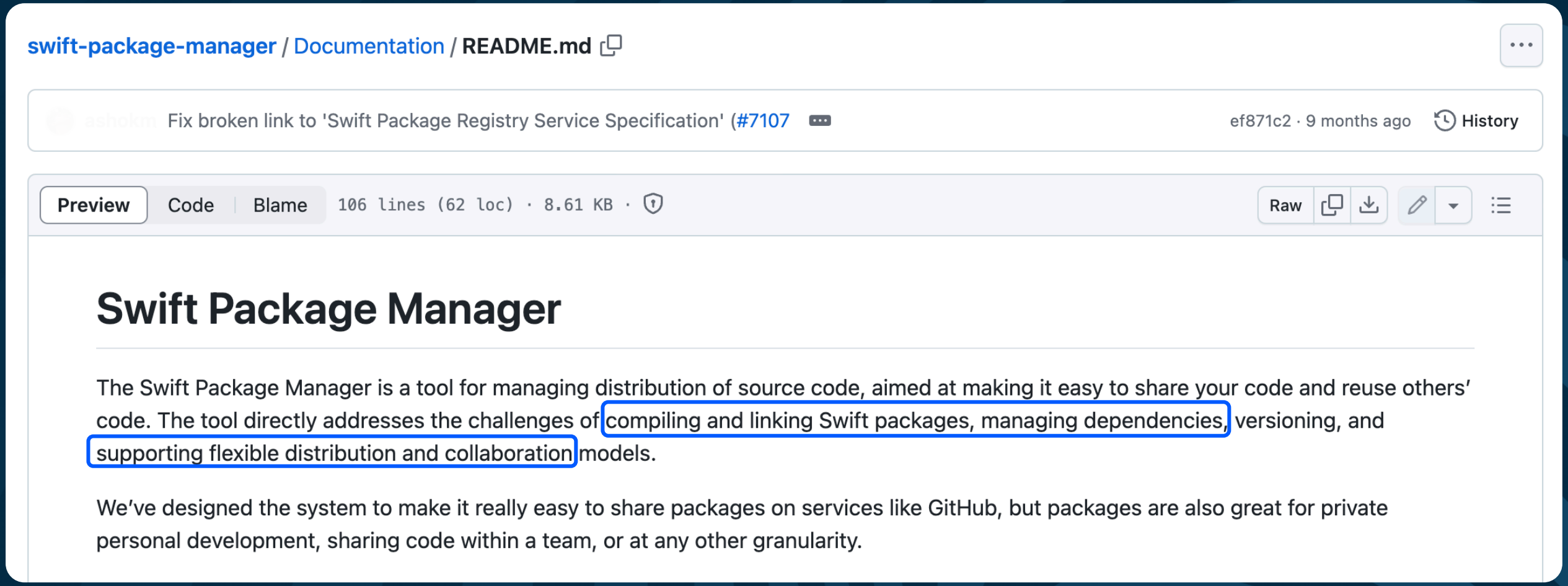
Все же уже знают, да?

- SPM — уже привычный для iOS разработчиков инструмент, которым можно и нужно пользоваться

- Минута молчания в честь падших в виде подов

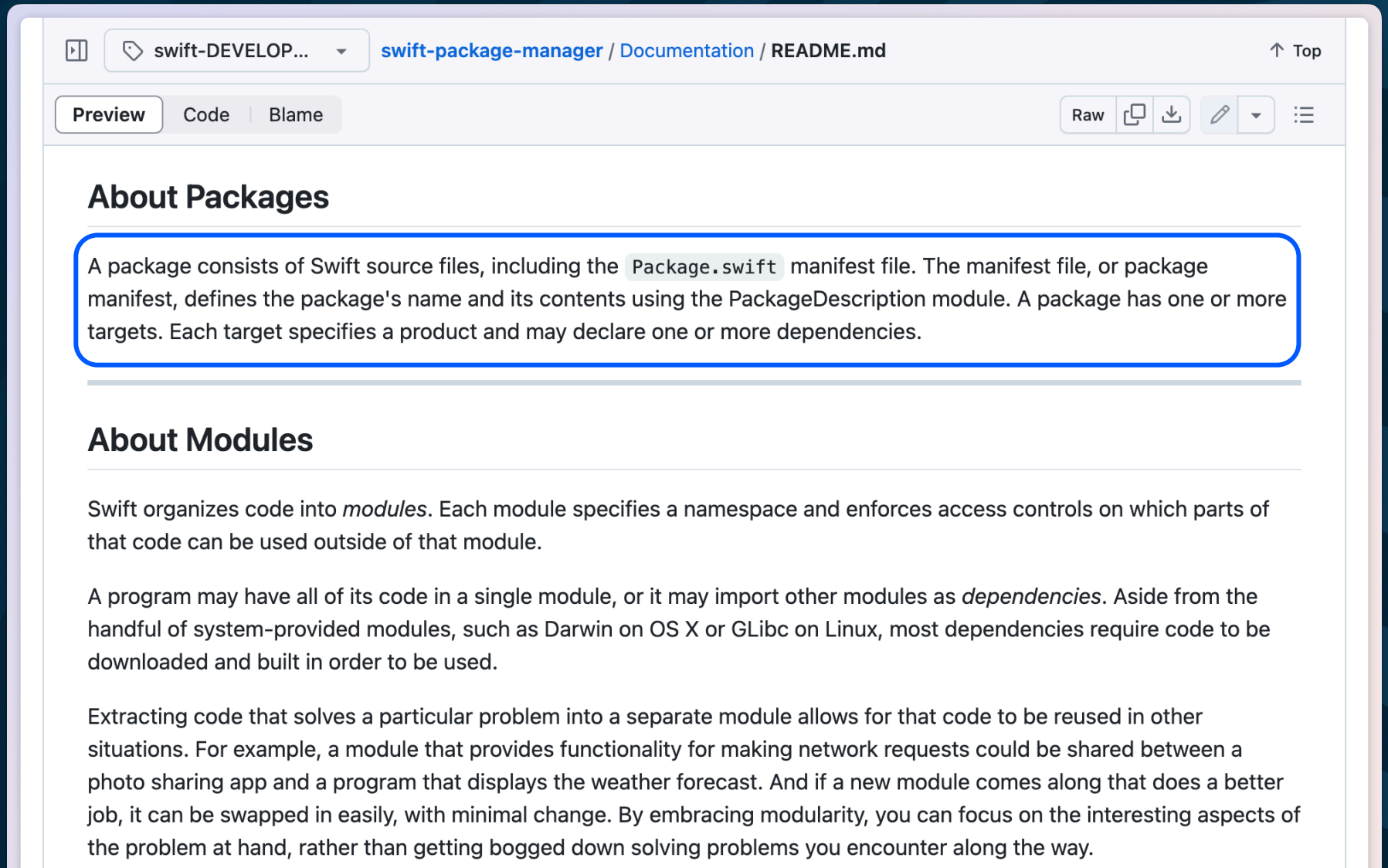
Что за Swift Package?

Все же уже знают, да?



Что за Swift Package?

Все же уже знают, да?



Что за Swift Package?

Какие-то зависимости и ресурсы. Фу

```swift
import PackageDescription

let package = Package(
    name: "MyPackage",
    dependencies: [
        .package(
            url: "https://github.com/apple/example-package-playingcard.git",
            from: "0.1.0"
        ),
    ]
)
```

За что Git?

И в чём его недостатки

```
~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4 (5m 24.15s)
swift package purge-cache; \
swift package reset; \
swift package resolve

Fetching                        ios_common/SnapKit.git
Fetching                        ios_common/KeychainAccess.git
Fetching                        ios_common/AppsFlyerFramework
Fetching                        sxm_ios/moya.git
Fetching                        ios_common/lottie-ios.git
Fetching                        ios_common/Alamofire.git
Fetched                        ios_common/SnapKit.git from cache (5.7
0s)
Fetching                        ios_common/needle.git
Fetched                        ios_common/needle.git from cache (126.
93s)
Fetched                        ios_common/AppsFlyerFramework from cac
he (313.69s)
Fetched                        ios_common/Alamofire.git from cache (3
13.69s)
Fetched                        ios_common/lottie-ios.git from cache (
313.69s)
Fetched                        sxm_ios/moya.git from cache (313.69s)
Fetched                        ios_common/KeychainAccess.git from cac
he (313.69s)
Computing version for                       ios_common/SnapKit.git
Computed                        ios_common/SnapKit.git at 5.0.1 (1.19
s)
Computing version for                       sxm_ios/moya.git
Computed                        sxm_ios/moya.git at 15.0.2 (0.49s)
Computing version for                       ios_common/lottie-ios.gi
t
Computed                        ios_common/lottie-ios.git at 4.4.2 (0
.55s)
```

За что Git?

И в чём его недостатки

```
~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4 (5m 24.15s)
swift package purge-cache; \
swift package reset; \
swift package resolve

Fetching                        ios_common/SnapKit.git
Fetching                        ios_common/KeychainAccess.git
Fetching                        ios_common/AppsFlyerFramework
Fetching                        sxm_ios/moya.git
Fetching                        ios_common/lottie-ios.git
Fetching                        ios_common/Alamofire.git
Fetched                        ios_common/SnapKit.git from cache (5.7
0s)
Fetching                        ios_common/needle.git
Fetched                        ios_common/needle.git from cache (126.
93s)
Fetched                        ios_common/AppsFlyerFramework from cac
he (313.69s)
Fetched                        ios_common/Alamofire.git from cache (3
13.69s)
Fetched                        ios_common/lottie-ios.git from cache (
313.69s)
Fetched                        sxm_ios/moya.git from cache (313.69s)
Fetched                        ios_common/KeychainAccess.git from cac
he (313.69s)
Computing version for                       ios_common/SnapKit.git
Computed                        ios_common/SnapKit.git at 5.0.1 (1.19
s)
Computing version for                       sxm_ios/moya.git
Computed                        sxm_ios/moya.git at 15.0.2 (0.49s)
Computing version for                       ios_common/lottie-ios.gi
t
Computed                        ios_common/lottie-ios.git at 4.4.2 (0
.55s)
```

За что Git?

И в чём его недостатки

```
.package(
    url: "https://github.com/Moya/Moya.git",
    exact: "15.0.3"
),
// ... depends on
.package(
    url: "https://github.com/Alamofire/Alamofire.git",
    exact: "5.6.2"
),
```

За что Git?

И в чём его недостатки

```
.package(
    url: "https://github.com/Moya/Moya.git",
    exact: "15.0.3"
),
// ... depends on
.package(
    url: "https://github.com/Alamofire/Alamofire.git",
    exact: "5.6.2"
),
```

```
public struct GitRepositoryProvider: RepositoryProvider {

    // ...

    public func fetch(
        repository: RepositorySpecifier,
        to path: Basics.AbsolutePath,
        progressHandler: FetchProgress.Handler? = nil
    ) throws {
        try self.clone(
            repository,
            repository.location.gitURL,
            path.pathString,
            ["--mirror"],
            progress: progressHandler
        )
    }
}
```

За что Git?

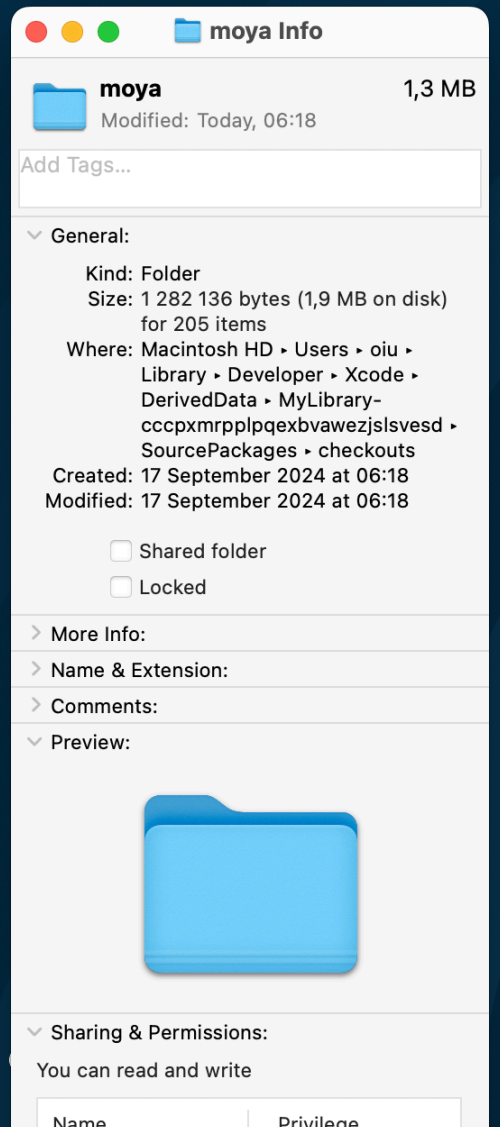
И в чём его недостатки

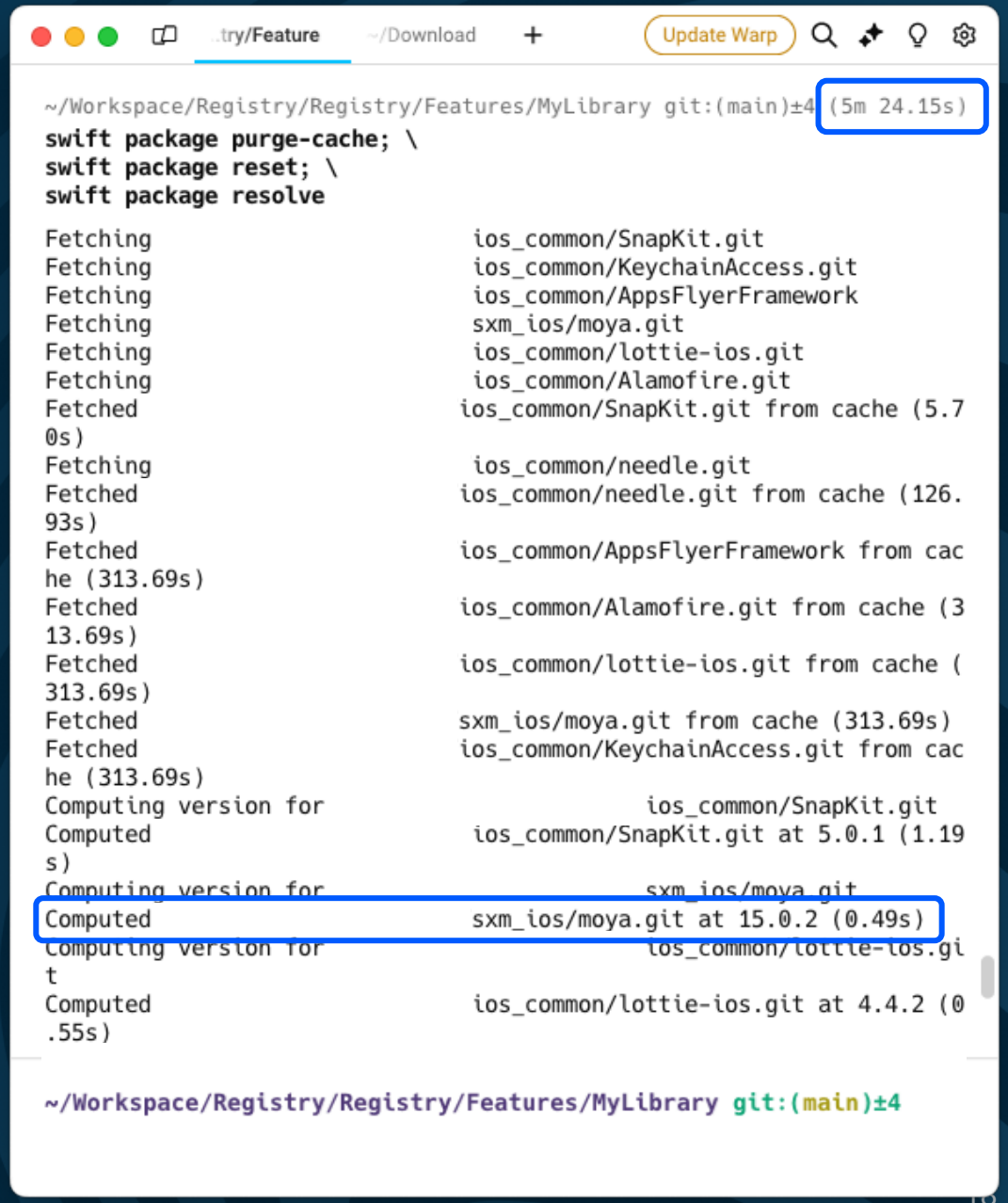
```
1 .package(
2     url: "https://github.com/Moya/Moya.git",
3     exact: "15.0.3"
4 ),
5 // ... depends on
6 .package(
7     url: "https://github.com/Alamofire/Alamofire.git",
8     exact: "5.6.2"
9 ),
```

```
~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4 (5m 24.15s)
swift package purge-cache; \
swift package reset; \
swift package resolve

Fetching                          ios_common/SnapKit.git
Fetching                          ios_common/KeychainAccess.git
Fetching                          ios_common/AppsFlyerFramework
Fetching                          sxm_ios/moya.git
Fetching                          ios_common/lottie-ios.git
Fetching                          ios_common/Alamofire.git
Fetched                           ios_common/SnapKit.git from cache (5.7
0s)
Fetching                          ios_common/needle.git
Fetched                           ios_common/needle.git from cache (126.
93s)
Fetched                           ios_common/AppsFlyerFramework from cac
he (313.69s)
Fetched                           ios_common/Alamofire.git from cache (3
13.69s)
Fetched                           ios_common/lottie-ios.git from cache (
313.69s)
Fetched                           sxm_ios/moya.git from cache (313.69s)
Fetched                           ios_common/KeychainAccess.git from cac
he (313.69s)
Computing version for                          ios_common/SnapKit.git
Computed                          ios_common/SnapKit.git at 5.0.1 (1.19
s)
Computing version for                          sxm_ios/moya.git
Computed                          sxm_ios/moya.git at 15.0.2 (0.49s)
Computing version for                          ios_common/lottie-ios.gi
t
Computed                          ios_common/lottie-ios.git at 4.4.2 (0
.55s)

~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4
```

За что Git?

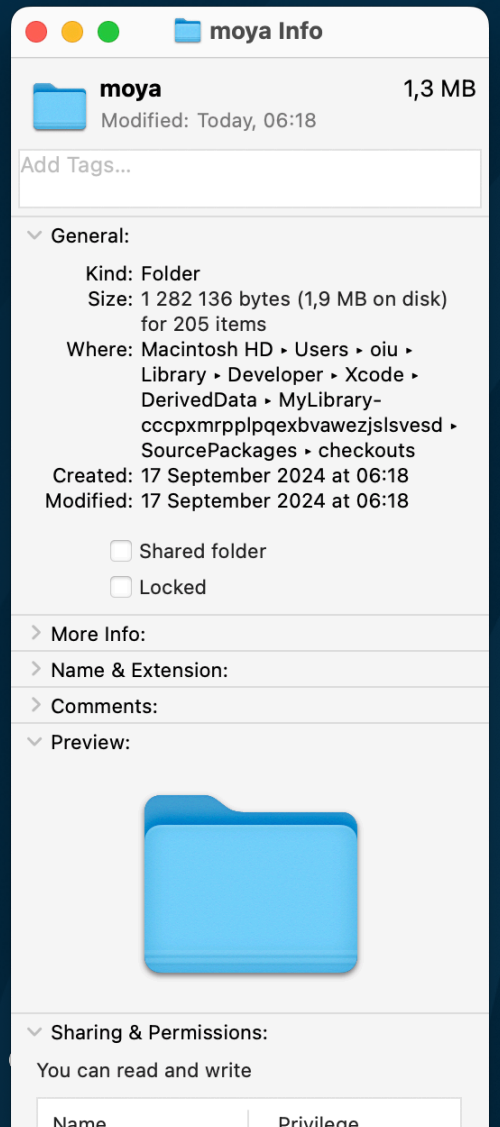
И в чём его недостатки

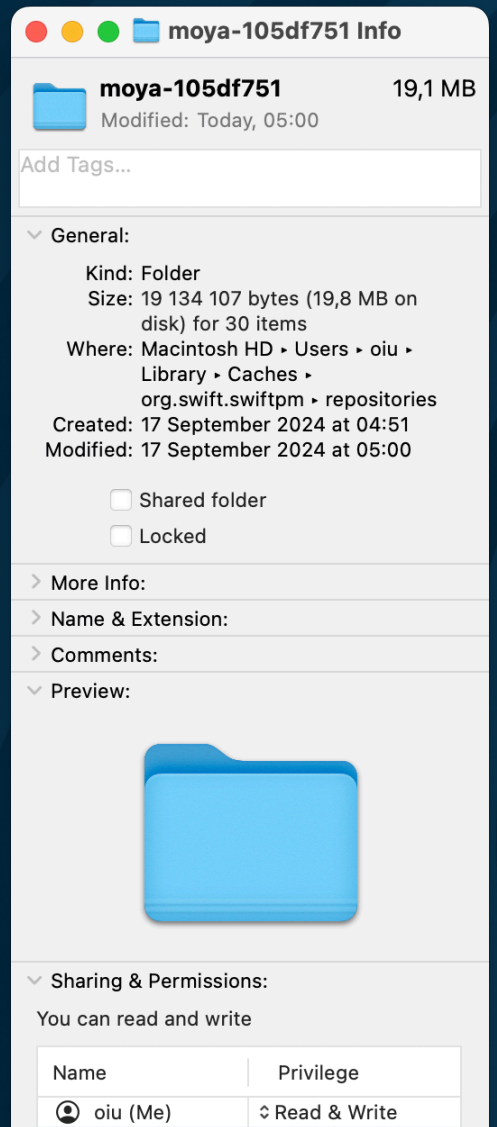


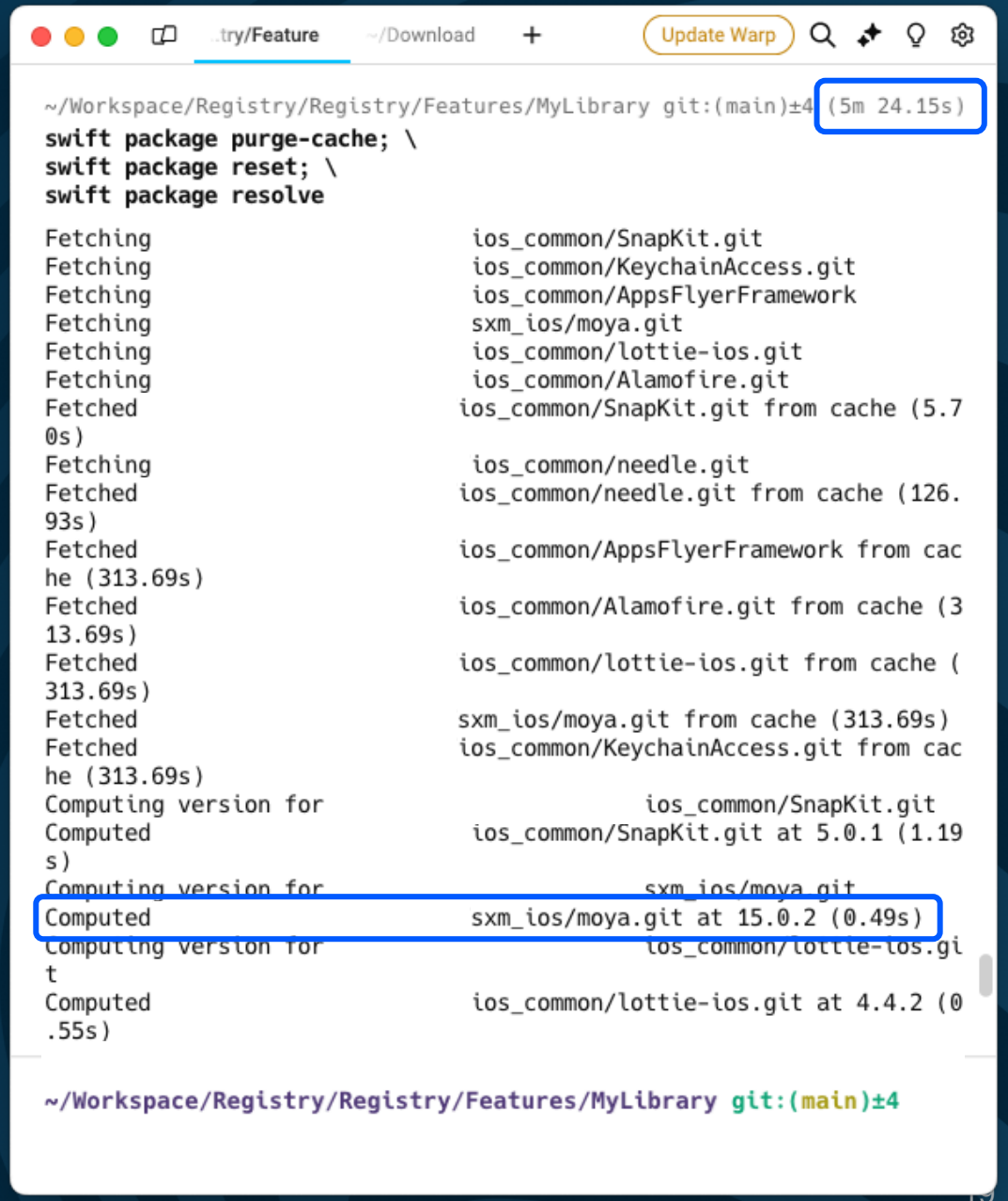


За что Git?

И в чём его недостатки

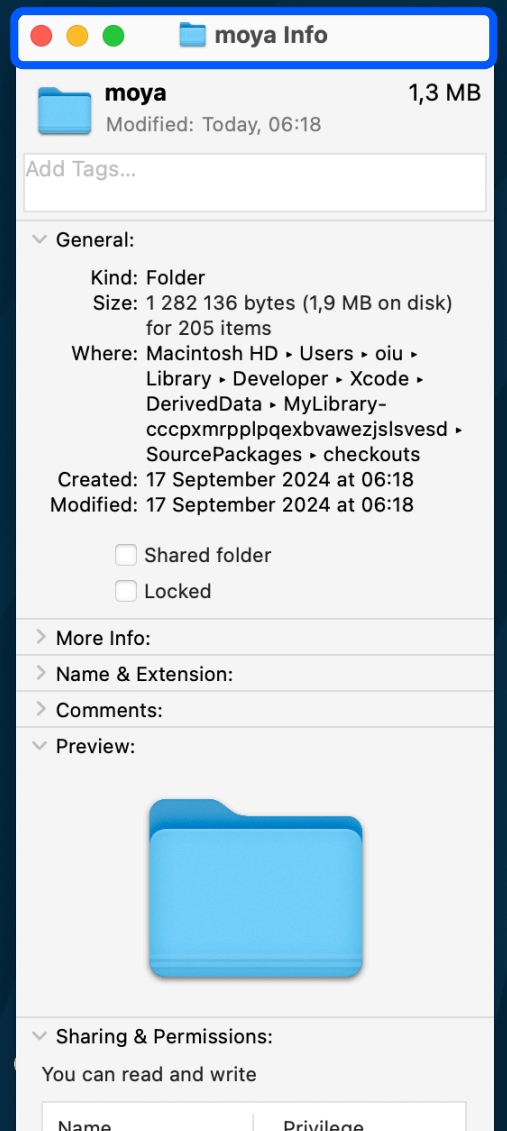


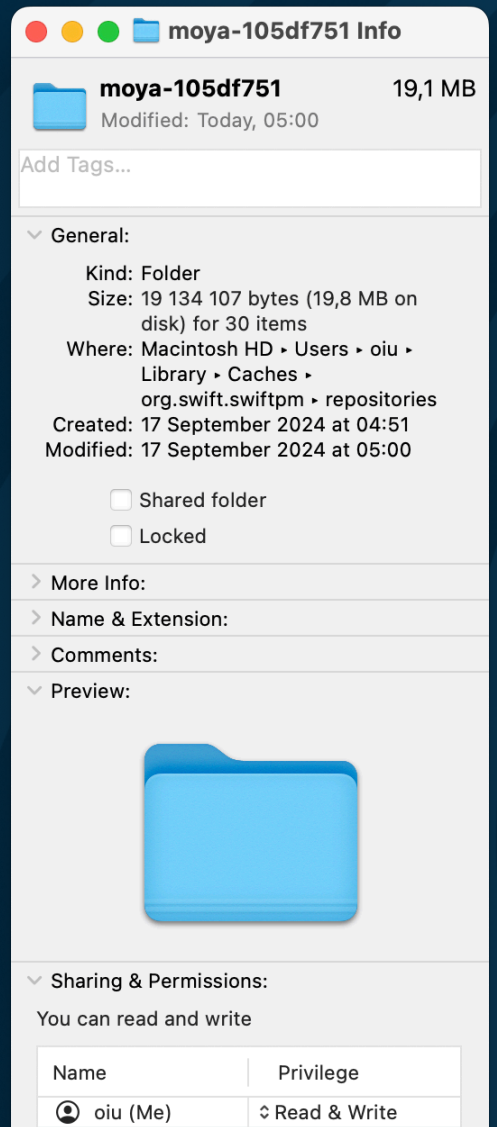


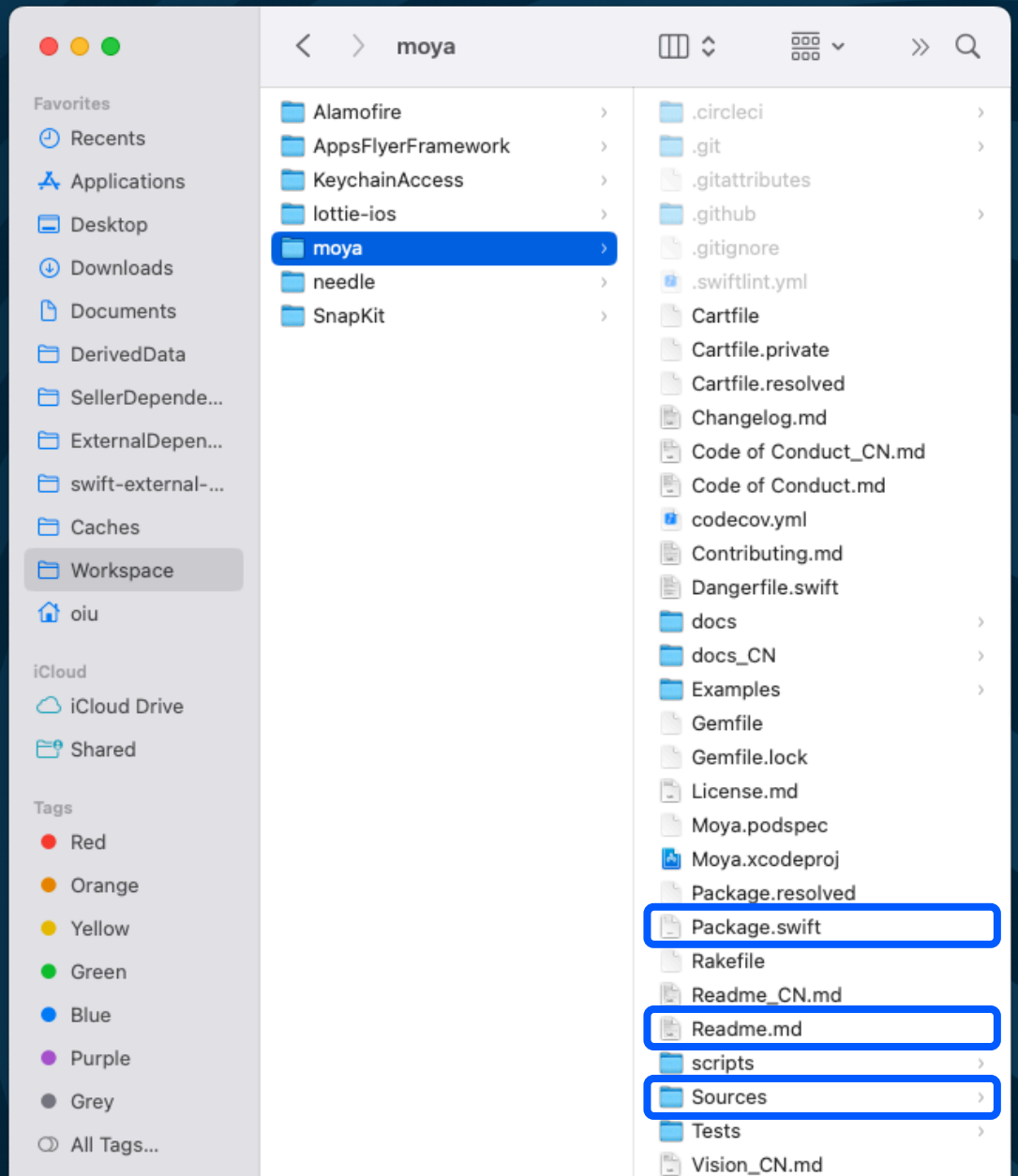


За что Git?

И в чём его недостатки

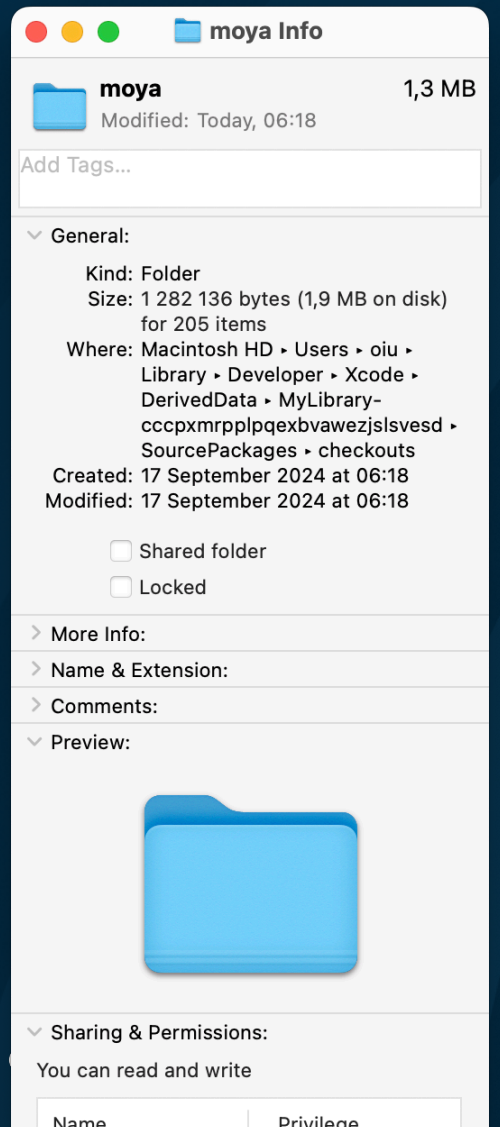


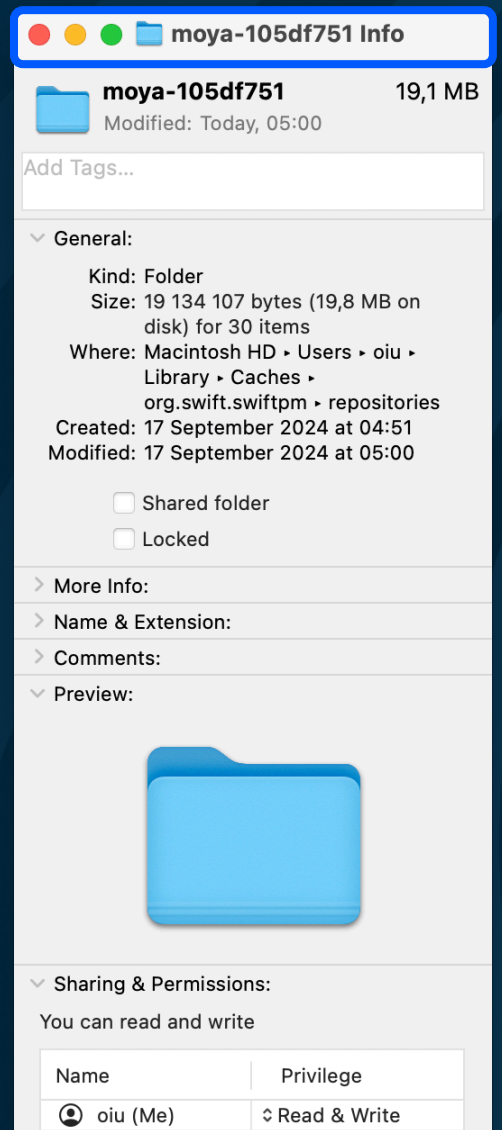


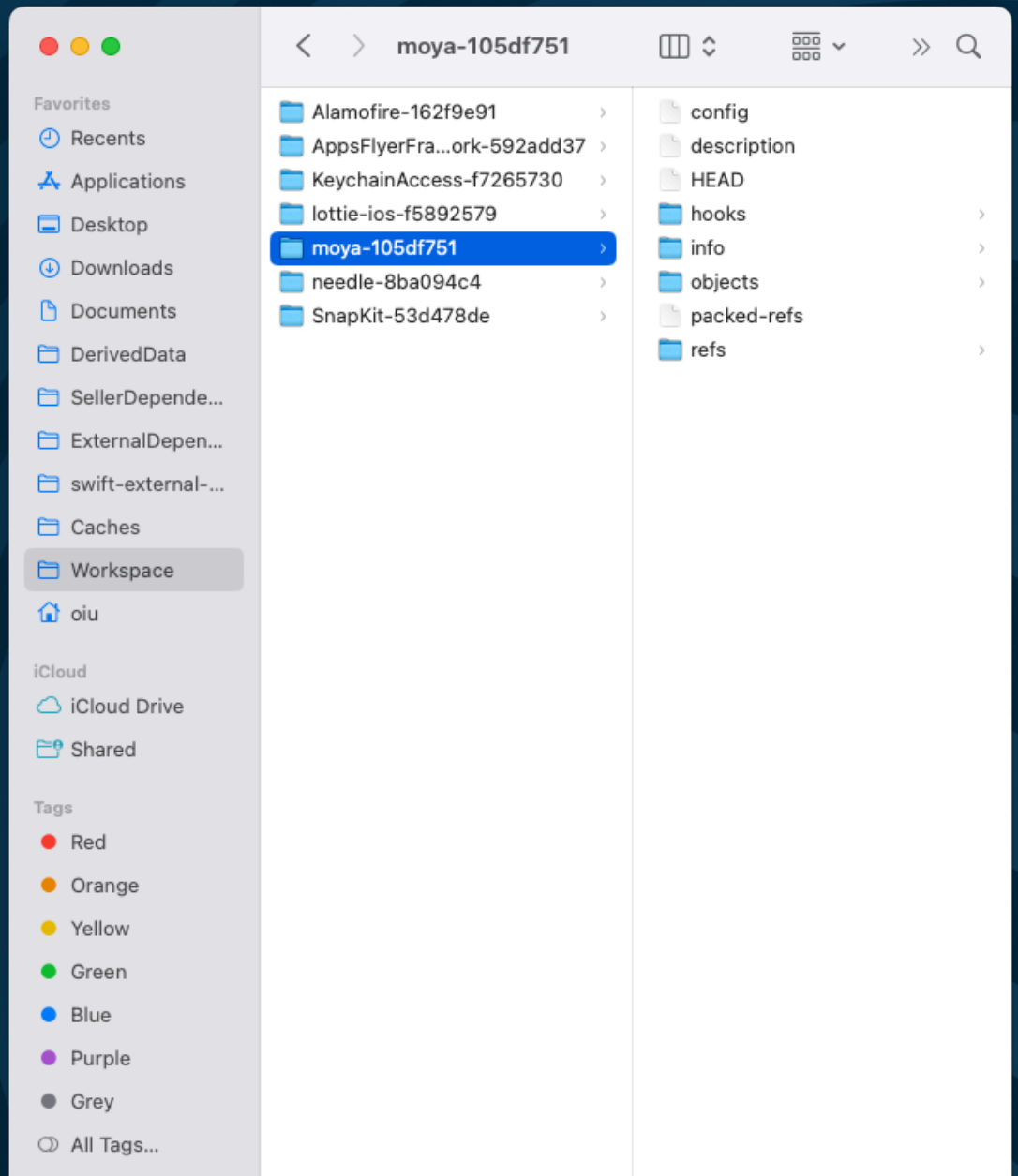


За что Git?

И в чём его недостатки

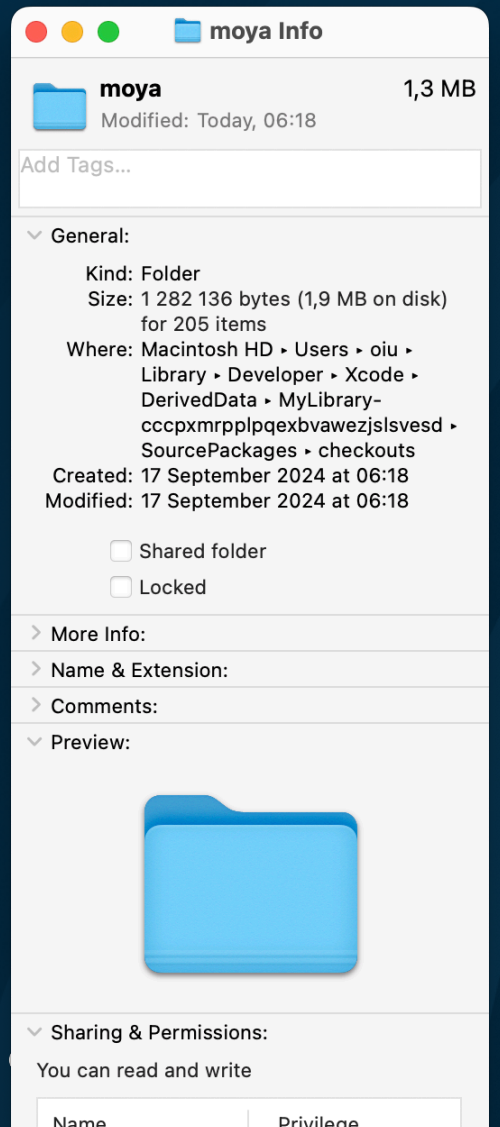


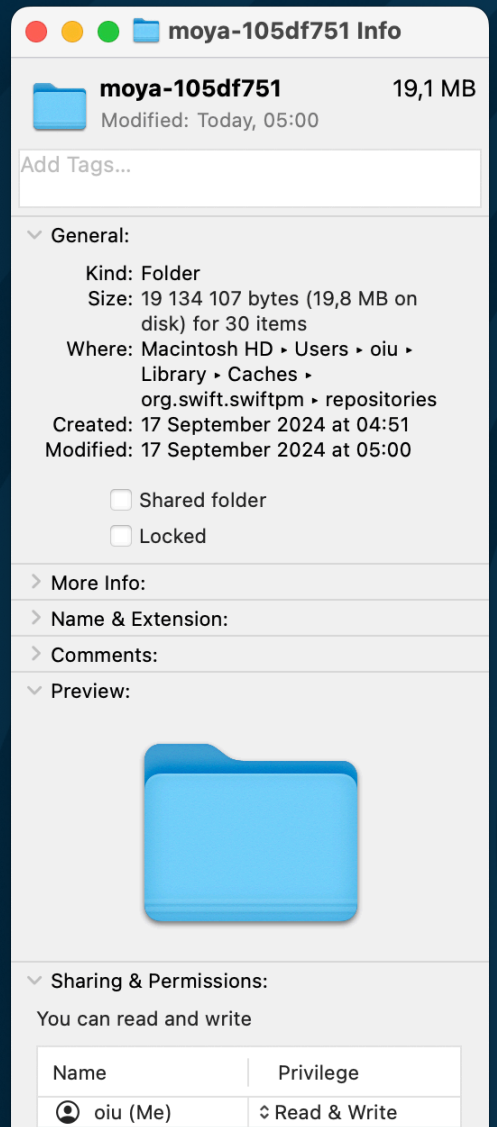


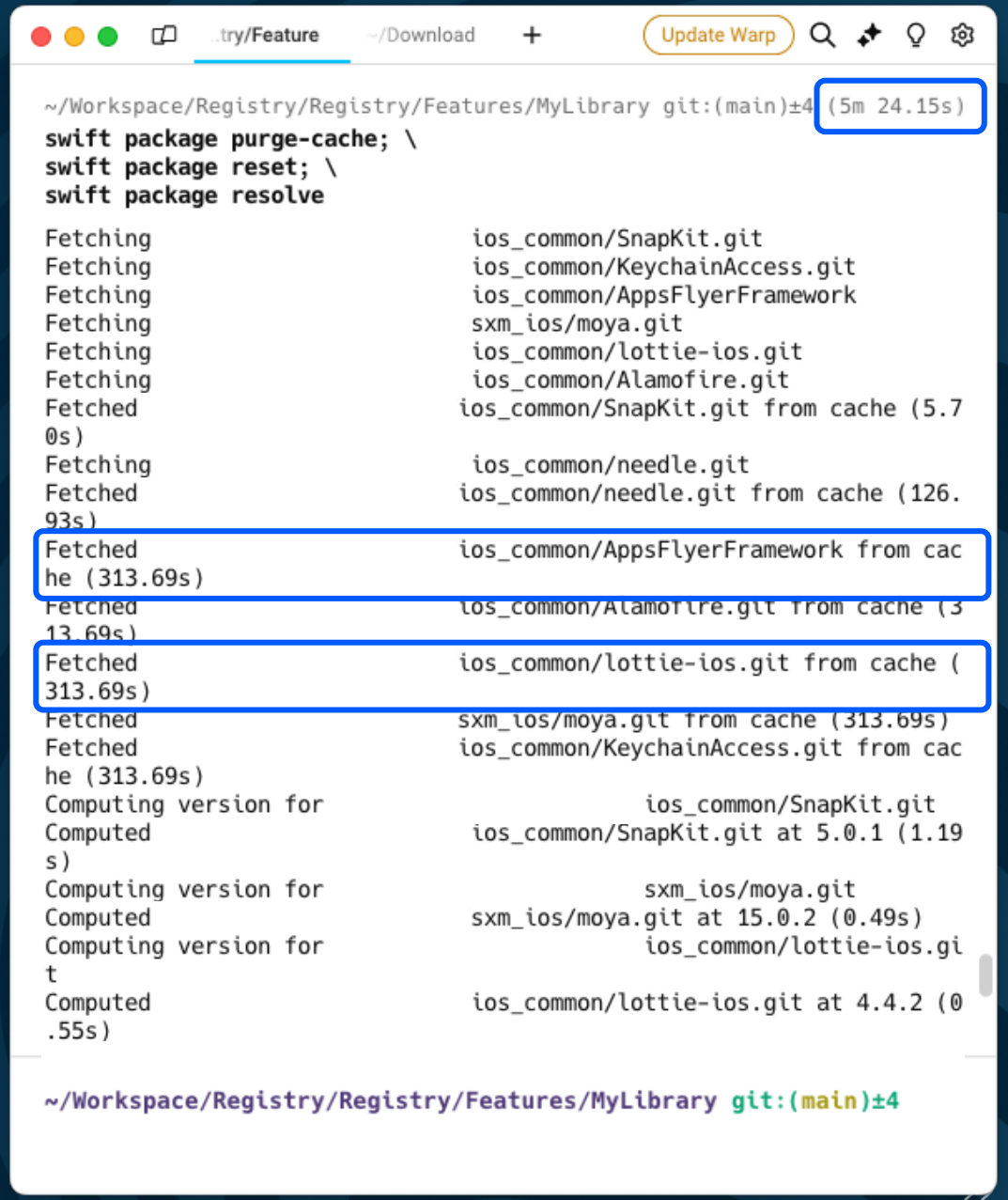


За что Git?

И в чём его недостатки

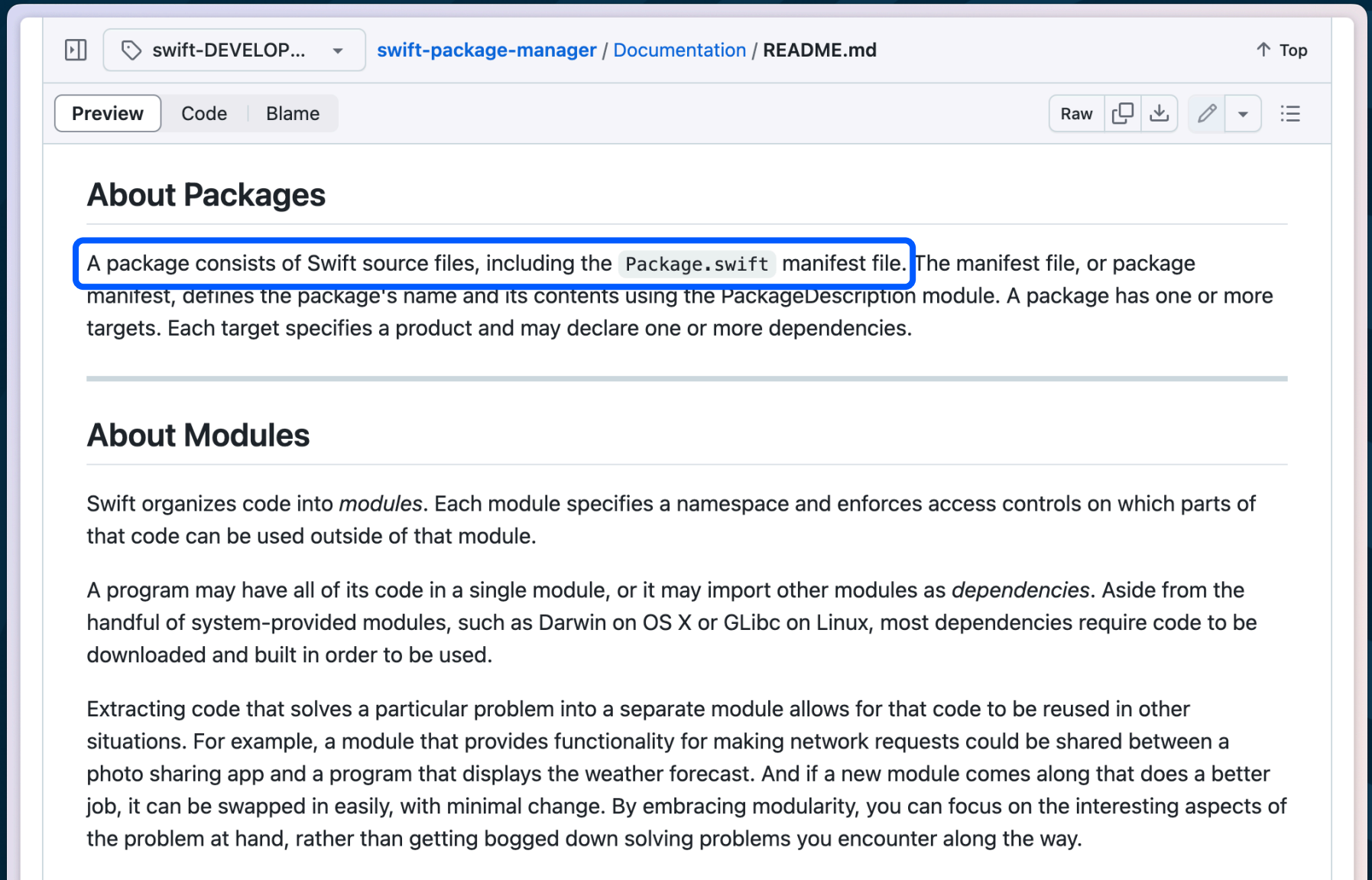






Откуда Git?

И за что



Откуда Git?

И за что

0.1.0 swift-package-manager / Documentation / Usage.md ↑ Top

Preview Code Blame Raw

Create a Package

Simply put: a package is a git repository with semantically versioned tags, that contains Swift sources and a `Package.swift` manifest file at its root.

Create a library package

A library package contains code which other packages can use and depend on. To get started, create a directory and run `swift package init` command:

```
$ mkdir MyPackage
$ cd MyPackage
$ swift package init # or swift package init --type library
$ swift build
$ swift test
```

This will create the directory structure needed for a library package with a target and the corresponding test target to write unit tests. A library package can contain multiple targets as explained in Target Format Reference.

Create an executable package

SwiftPM can create native binary which can be executed from command line. To get started:

Как избавиться от Git?

Что за Registry Service?

Preview | Code | Blame — Raw

Package Registry Service

- Proposal: SE-0292
- Authors: Bryan Clark, Whitney Imura, Mattt Zmuda
- Review Manager: Tom Doron
- Status: **Implemented (Swift 5.7)**
- Implementation: apple/swift-package-manager#3023
- Decision Notes: Rationale
- Review: 1 2 3 Amendment
- Previous Revision: 1 2 3

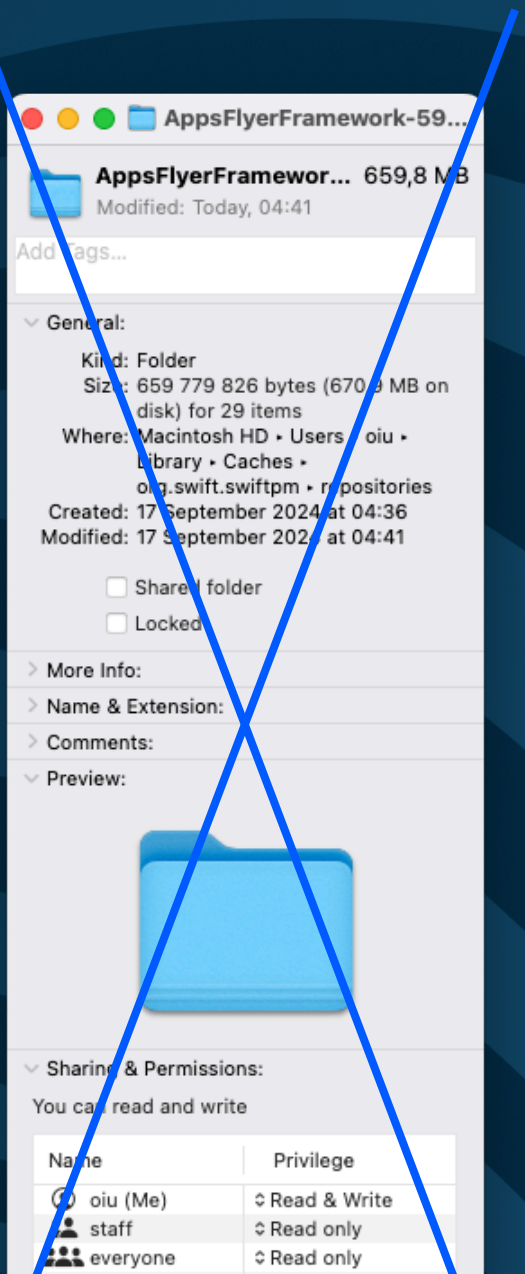
Introduction

Swift Package Manager downloads dependencies using Git. Our proposal defines a standard web service interface that it can also use to download dependencies from a package registry.

Swift-evolution thread: Swift Package Registry Service

Motivation

A package dependency is currently specified by a URL to its source repository. When Swift Package Manager builds a project for the first time, it clones the Git repository for each dependency and attempts to resolve the version requirements



02

Swift Registry Service

И как с ней работать

Что за Registry

В общем понимании

Кладём на место…

…забираем в том же виде

ozontech

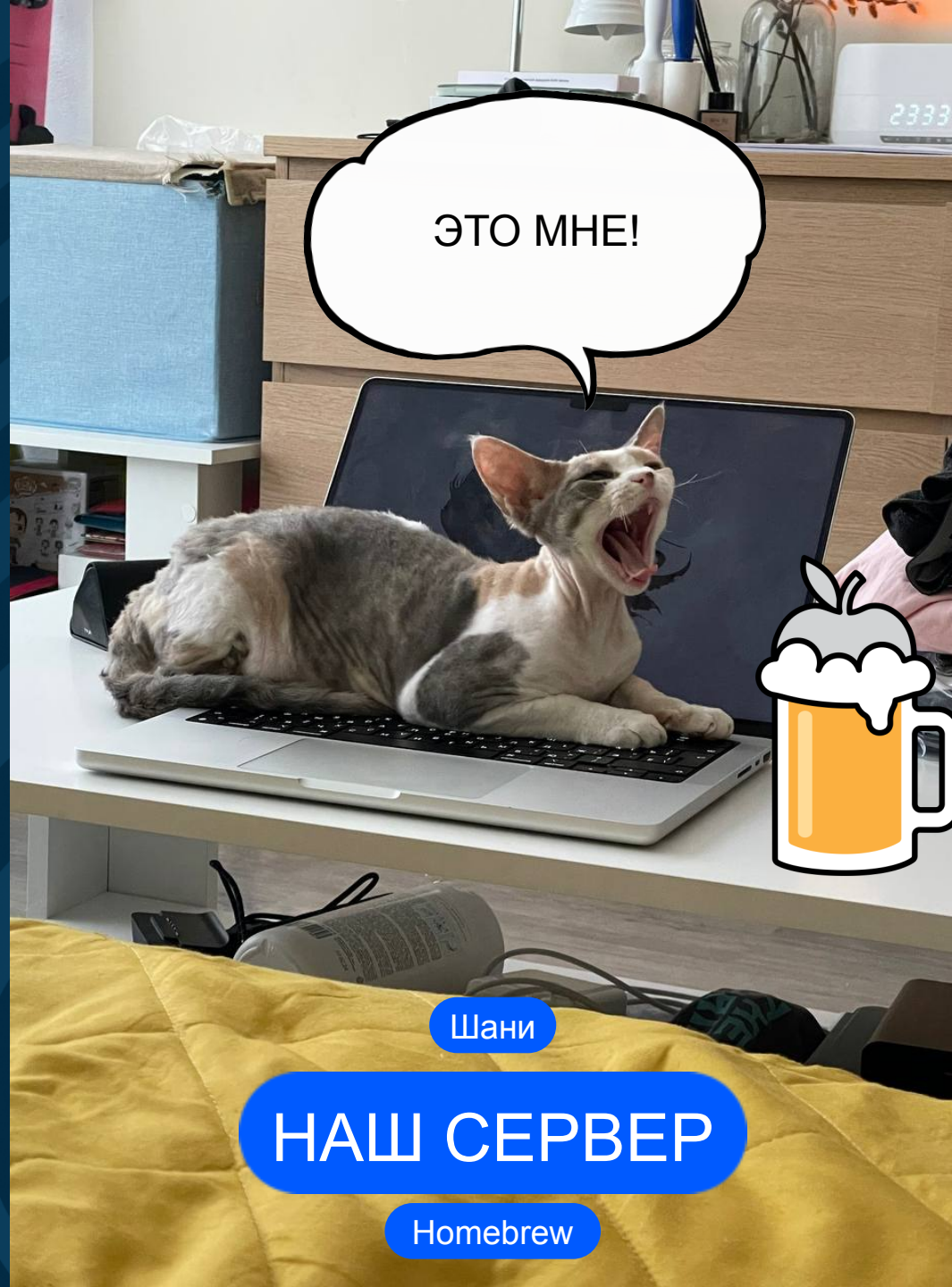
Что за Registry

В общем понимании

Кладём на место…

```
1 $ brew create foo.com/foo-1.0.tgz
2 Created /opt/homebrew/Library/Taps/homebrew/homebrew-
  core/Formula/foo.rb
```

ozontech

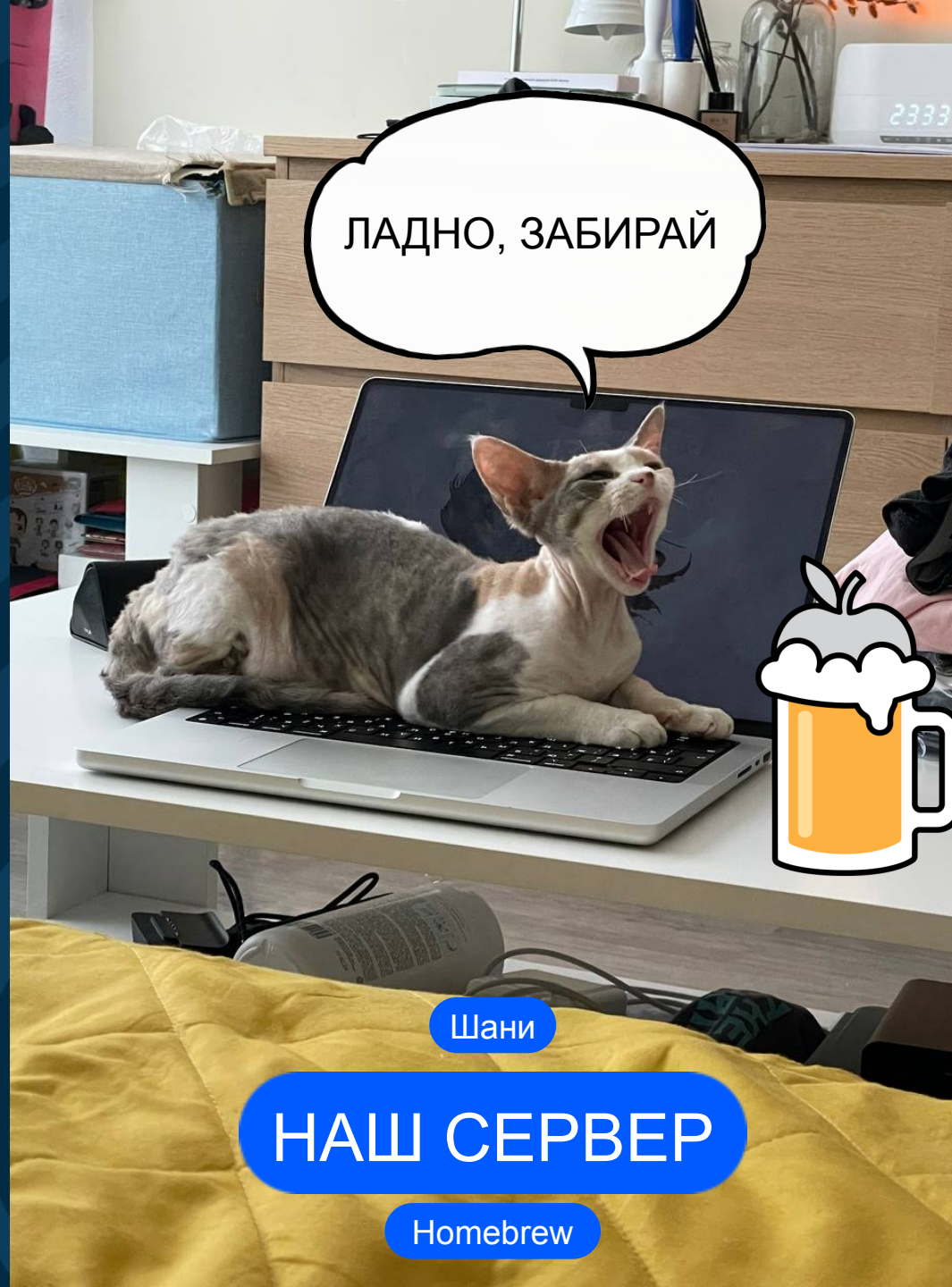


Что за Registry

В общем понимании

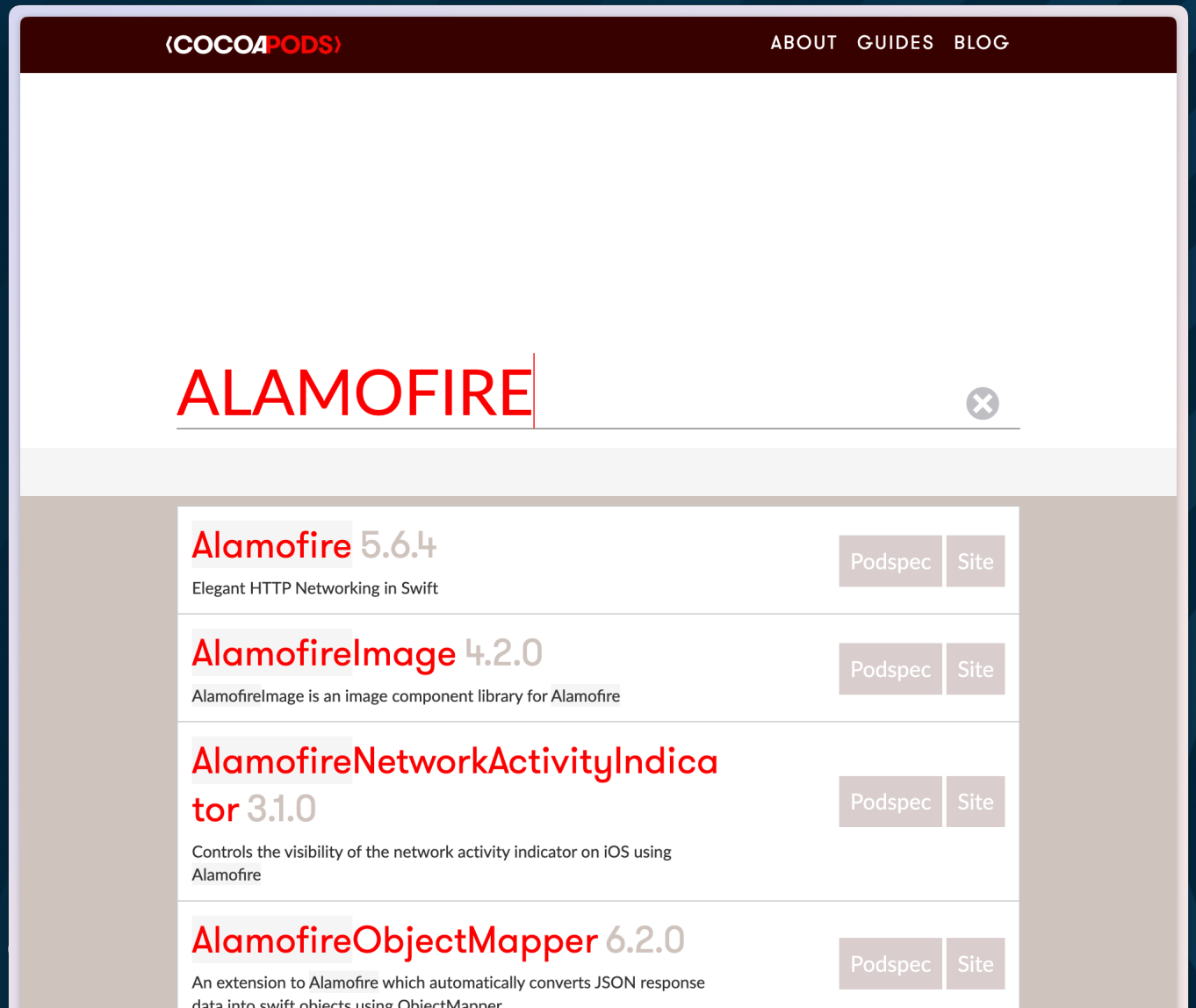
```
1 $ brew install foo
```

…забираем в том же виде



Что за Registry

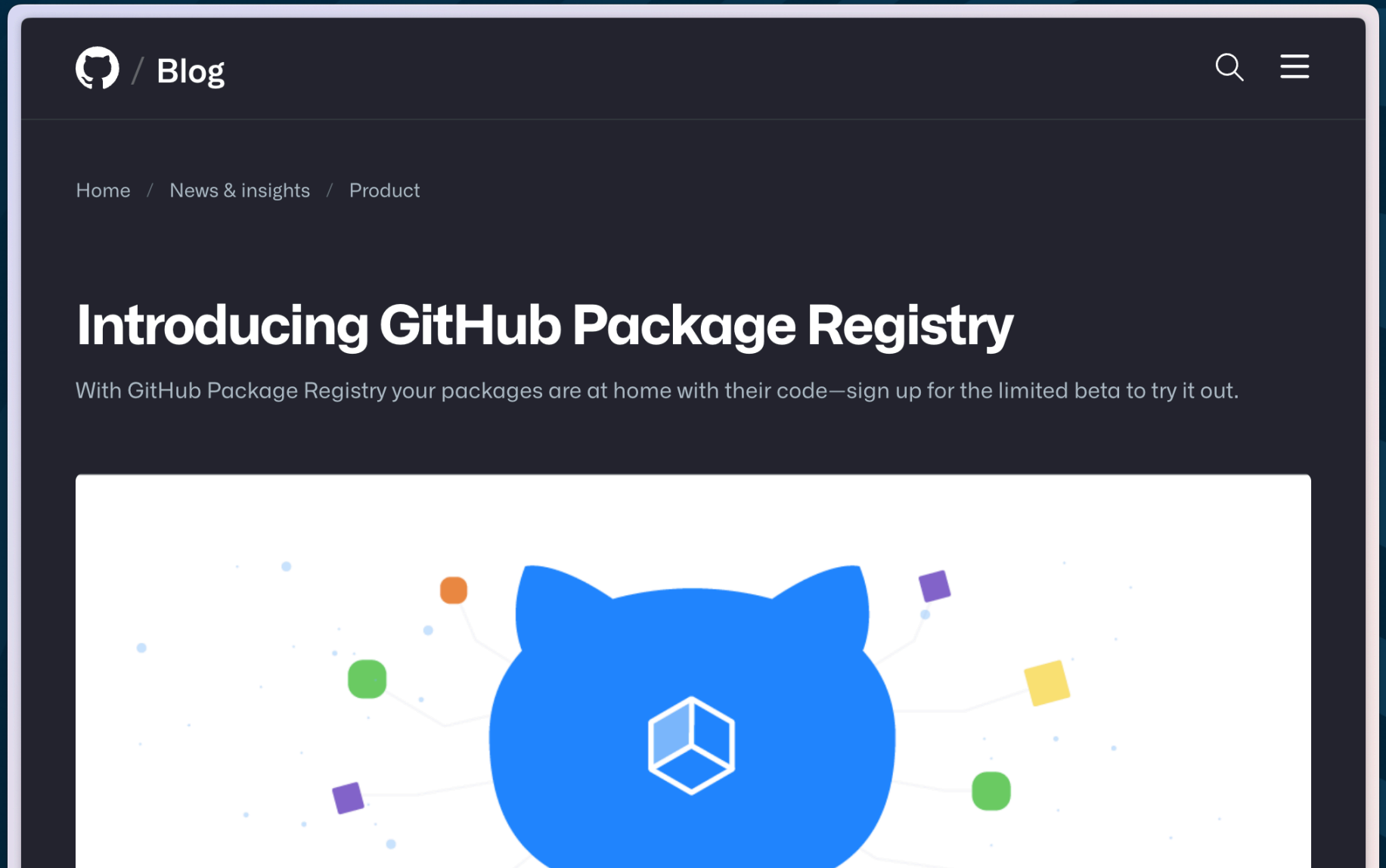
В понимании iOS




```
1 platform :ios, '8.0'
2 use_frameworks!
3
4 target 'MyApp' do
5   pod 'Alamofire', '~> 2.6'
6 end
```

Как он получился?

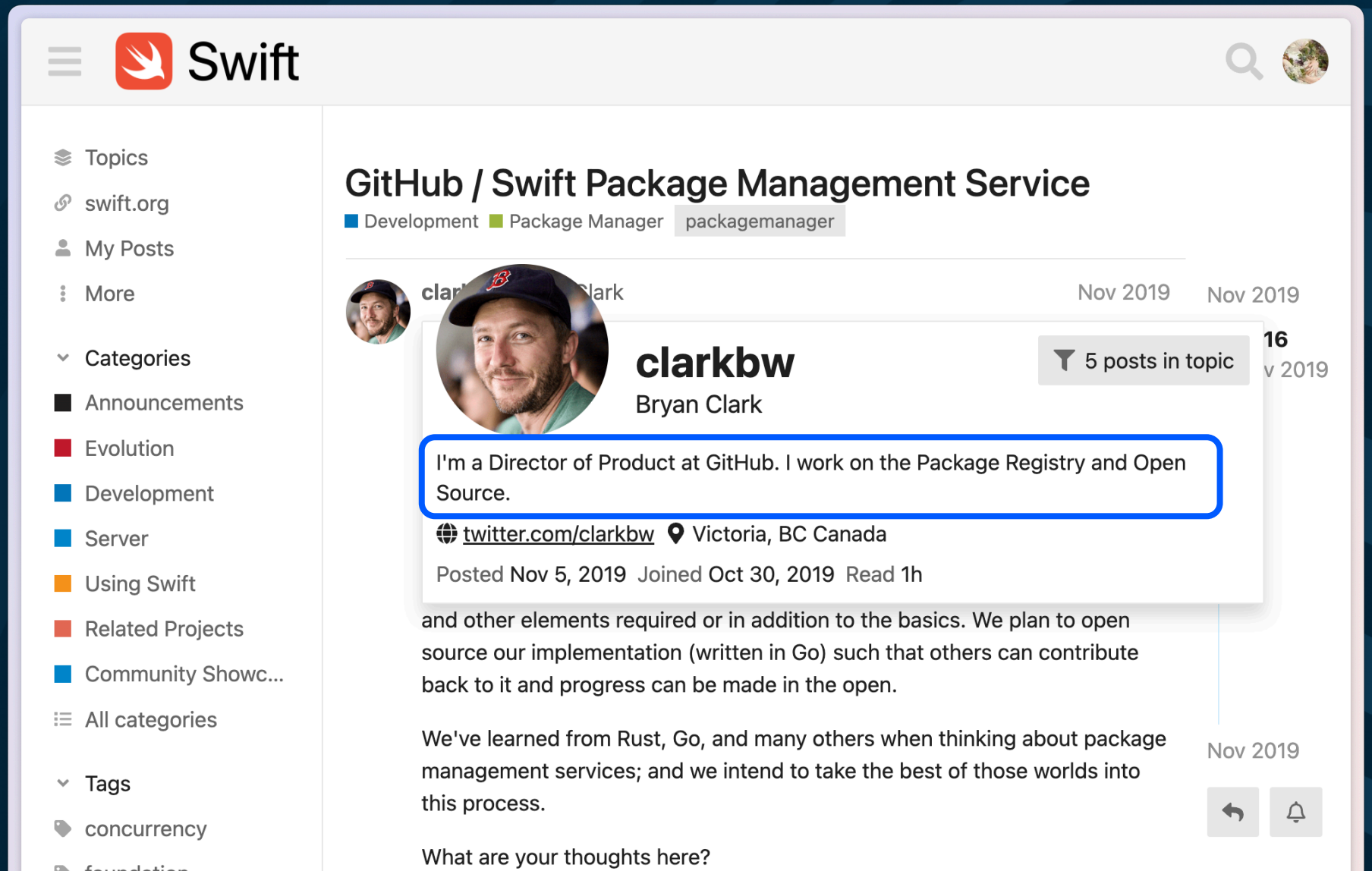
И почему вы о нём не слышите



ozontech

Как он получился?

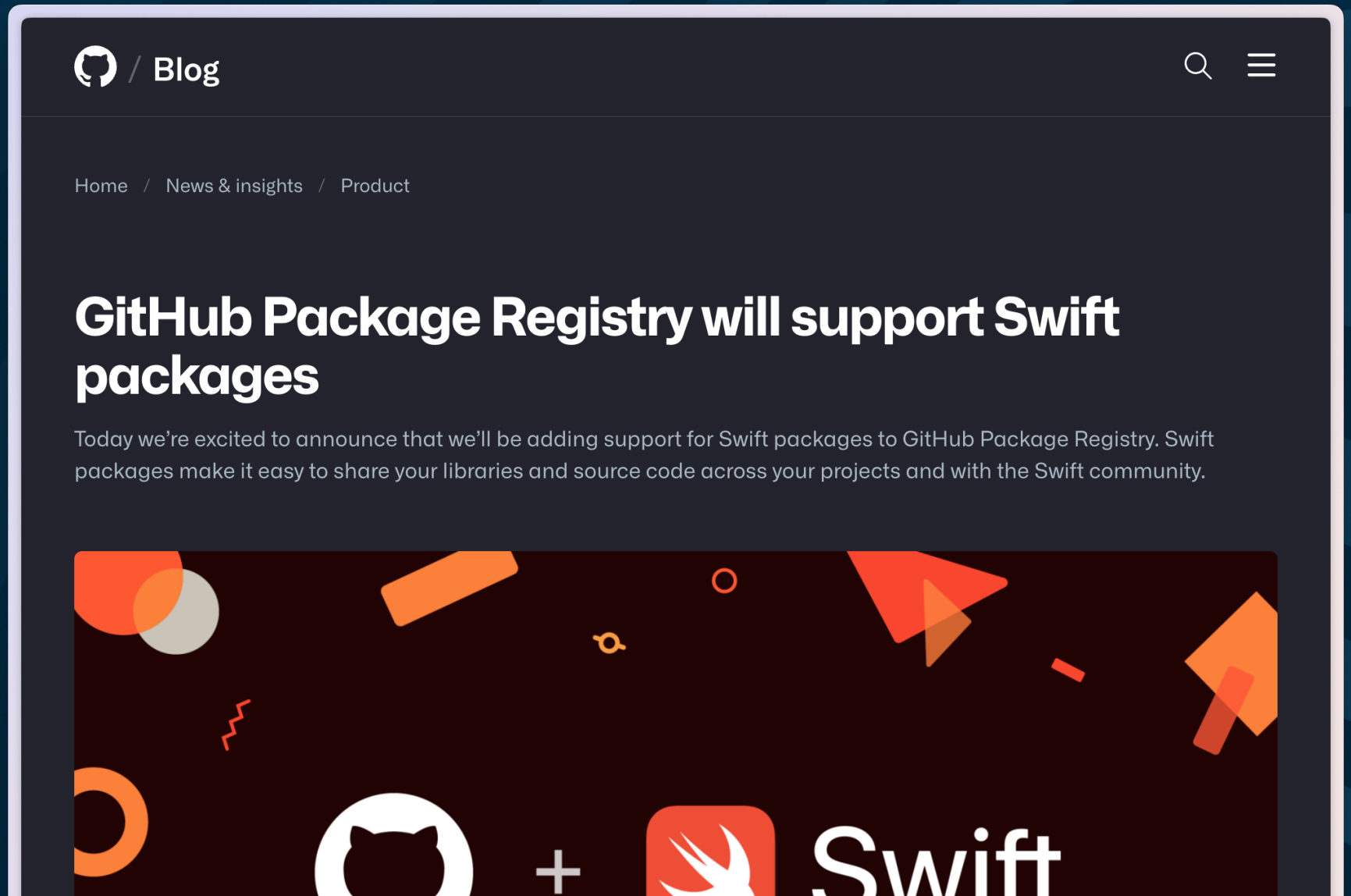
И почему вы о нём не слышите



ozontech

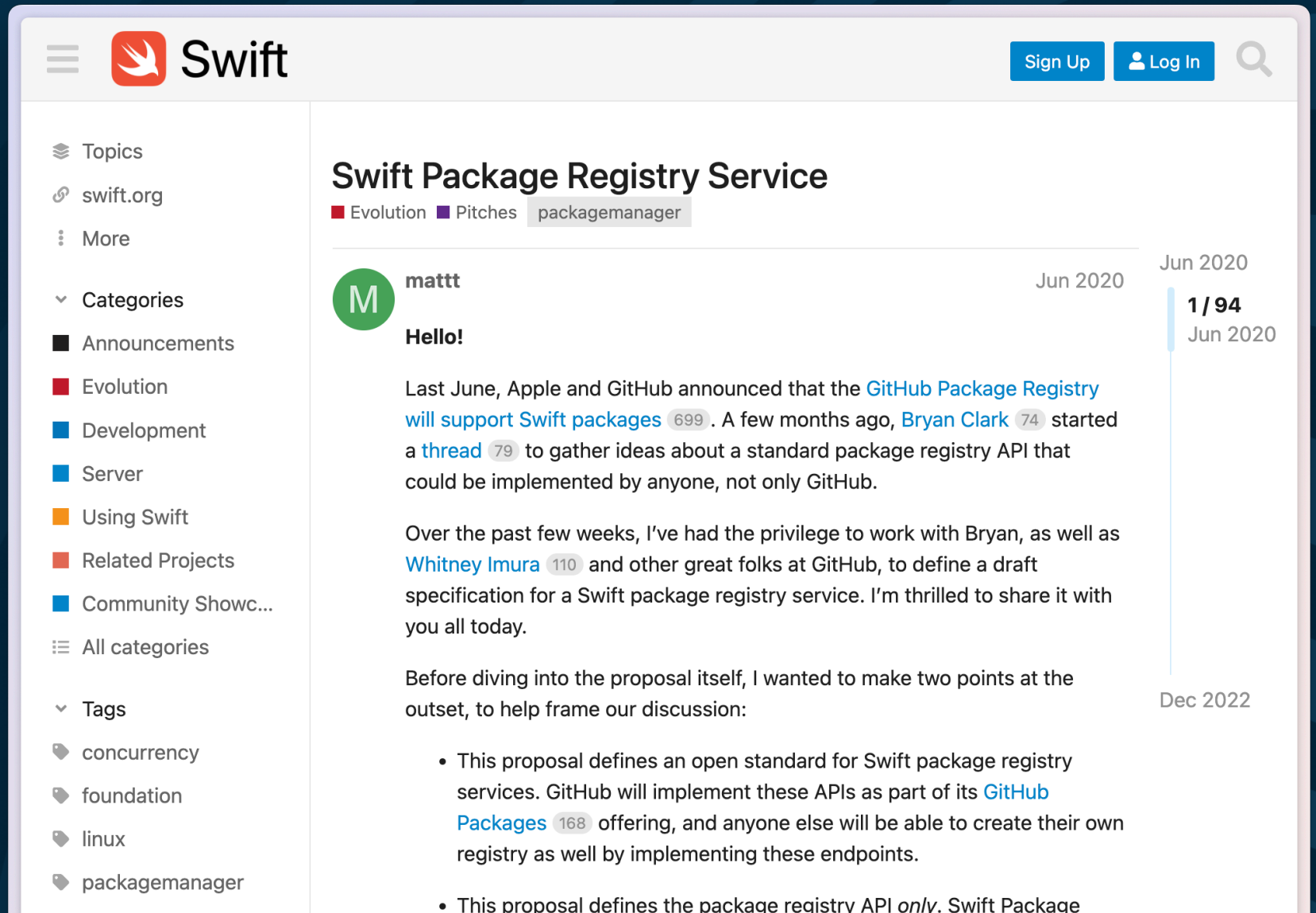
Как он получился?

И почему вы о нём не слышите



Как он получился?

И почему вы о нём не слышите



ozontech

Какие проблемы решает?

И откуда эти проблемы

Удобный способ распространения

- `swift package publish` вместо `git push`

Эффективность Git`a

- Консистентность
- Эффективность скачки и кэширования
- Эффективность просчёта версий

Общий индекс

- Поиск пакетов и их версий

Как убрали git

API Сервиса

Preview | Code | Blame — Raw

Package Registry Service

- Proposal: SE-0292
- Authors: Bryan Clark, Whitney Imura, Mattt Zmuda
- Review Manager: Tom Doron
- Status: **Implemented (Swift 5.7)**
- Implementation: apple/swift-package-manager#3023
- Decision Notes: Rationale
- Review: 1 2 3 Amendment
- Previous Revision: 1 2 3

Introduction

Swift Package Manager downloads dependencies using Git. Our proposal defines a standard web service interface that it can also use to download dependencies from a package registry.

Swift-evolution thread: Swift Package Registry Service

Motivation

A package dependency is currently specified by a URL to its source repository. When Swift Package Manager builds a

ozontech

Как убрали git
API Сервиса

main swift-evolution / proposals / 0292-package-registry-service.md ↑ Top

Preview Code Blame Raw

A package registry service implements the following REST API endpoints for listing releases for a package, fetching information about a release, and downloading the source archive for a release:

Method	Path	Description
`GET`	`/{scope}/{name}`	List package releases
`GET`	`/{scope}/{name}/{version}`	Fetch metadata for a package release
`GET`	`/{scope}/{name}/{version}/Package.swift{?swift-version}`	Fetch manifest for a package release
`GET`	`/{scope}/{name}/{version}.zip`	Download source archive for a package release
`GET`	`/identifiers{?url}`	Lookup package identifiers registered for a URL

A formal specification for the package registry interface is provided alongside this proposal. In addition, an OpenAPI (v3) document and a reference implementation written in Swift are provided for the convenience of developers interested in building their own package registry.

Как убрали git
API Сервиса

main swift-evolution / proposals / 0292-package-registry-service.md ↑ Top

Preview Code Blame Raw

New `PackageDescription` API

The `Package.Dependency` type adds the following static method:

```
extension Package.Dependency {
    /// Adds a dependency on a package with the specified identifier
    /// that uses the provided version requirement.
    public static func package(
        id: String,
        _ requirement: Package.Dependency.VersionBasedRequirement
    ) -> Package.Dependency
}
```

These methods may be called in the `dependencies` field of a package manifest to declare one or more dependencies by their respective package identifier.

```
dependencies: [
   .package(id: "mona.LinkedList", .upToNextMinor(from: "1.1.0")),
   .package(id: "mona.RegEx", .exact("2.0.0"))
]
```

A package dependency declared with an identifier using this method may only specify a version-based requirement. `Package.Dependency.VersionBasedRequirement` is a new type that provides the same interface as `Package.Dependency.Requirement` for version-based requirements, but excluding branch-based and commit-based requirements.

ozontech

Как работает сервис

А как вкачать?

https://шани-емае.ру/

leet-scope/leet-library/leet-library-1.33.7.zip

[6]	PUT	/{scope}/{name}/{version}

Как работает сервис

А как вкачать?

https://шани-емае.ру/

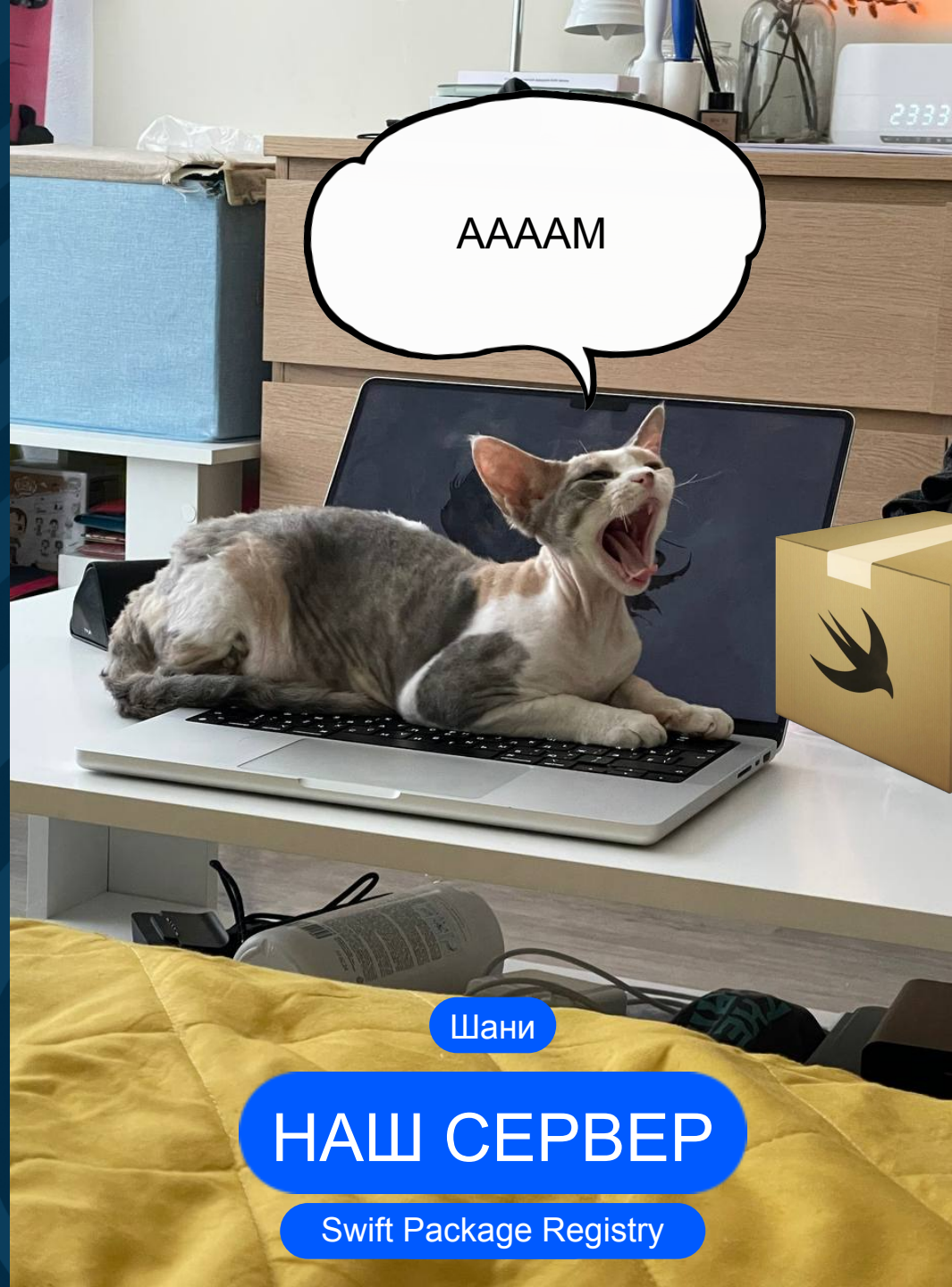
+

leet-scope/leet-library/leet-library-1.33.7.zip

[6]	PUT	/{scope}/{name}/{version}

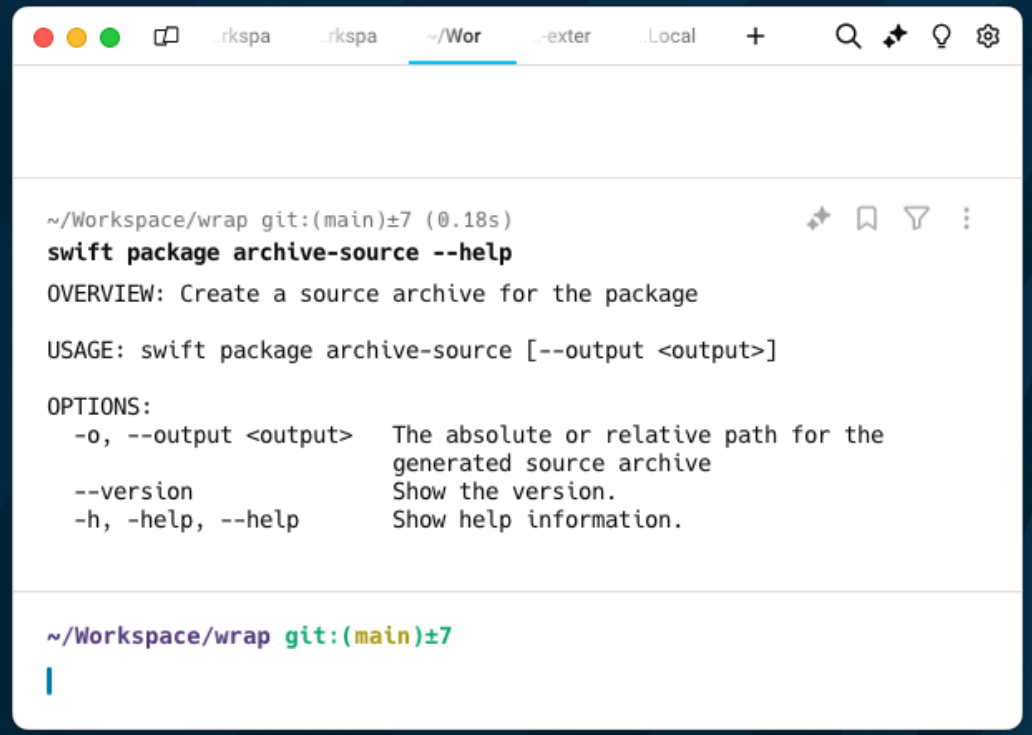
ozontech

шани/leet-scope/Leet-library-1.33.7.zip

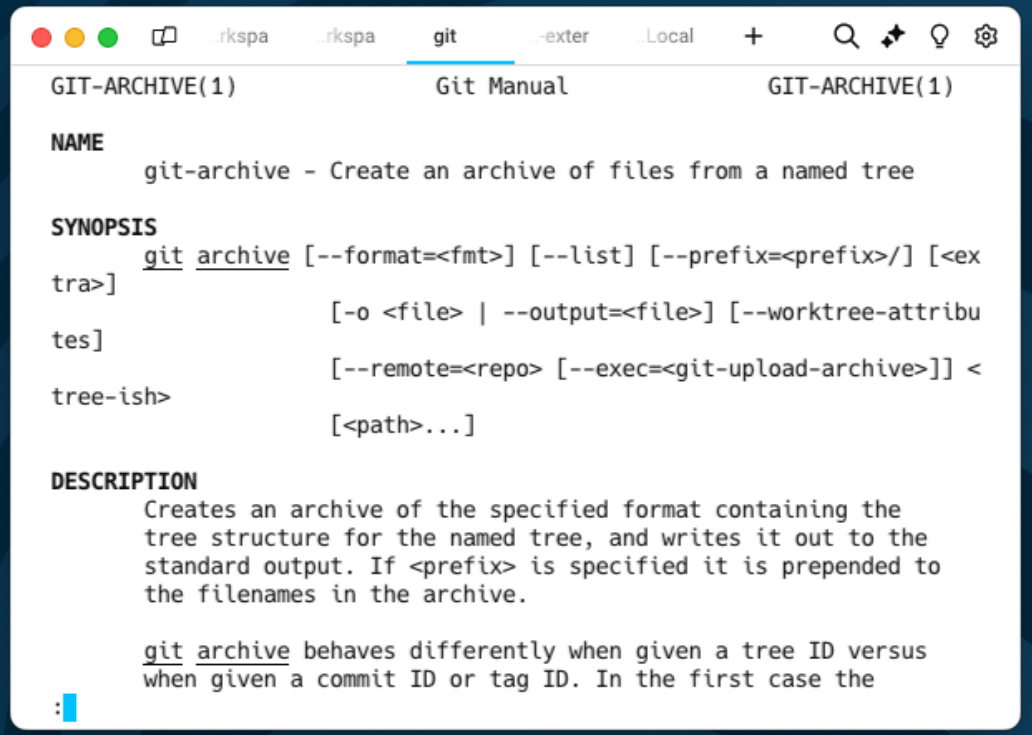


Как работает сервис

А как вкачать?



=



ozon tech

Как работает сервис

А как скачать?

https://шани-емае.ру/

leet-scope/leet-library/leet-library-1.33.7.zip

```
GET    /{scope}/{name}/{version}.zip
```

Как работает сервис

А как скачать?

https://шани-емае.ру/

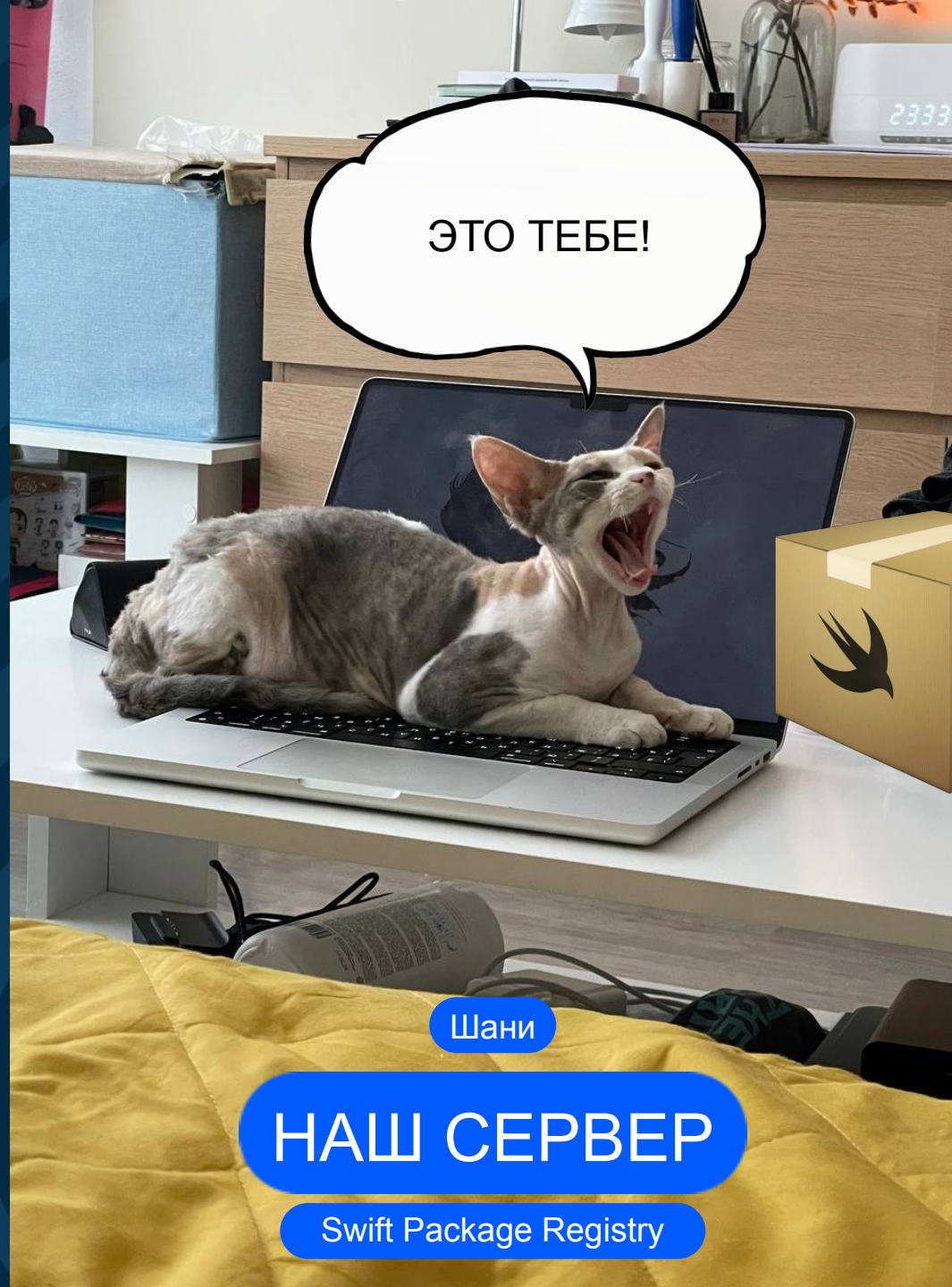
+

leet-scope/leet-library/leet-library-1.33.7.zip

```
GET    /{scope}/{name}/{version}.zip
```

build/registry/Leet-library
1.33.7.zip

ozontech



Как работает сервис

Как использовать у себя?

```swift
import PackageDescription

let package = Package(
    name: "MyPackage",
    dependencies: [
        .package(
            url: "https://github.com/Moya/Moya.git",
            from: "15.0.0"
        ),
    ]
)
```

```swift
import PackageDescription

let package = Package(
    name: "MyPackage",
    dependencies: [
        .package(
            id: "test.Moya",
            from: "15.0.0"
        ),
    ]
)
```

Как работает сервис

Как использовать у себя?

```
~/Workspace/swift-external-dependencies
swift package-registry set --help
OVERVIEW: Set a custom registry

USAGE: swift package-registry set [--global] [--scope <scope>] <url>

ARGUMENTS:
  <url>                   The registry URL

OPTIONS:
  --global                Apply settings to all projects for this user
  --scope <scope>         Associate the registry with a given scope
  --version               Show the version.
  -h, -help, --help       Show help information.

~/Workspace/swift-external-dependencies
```

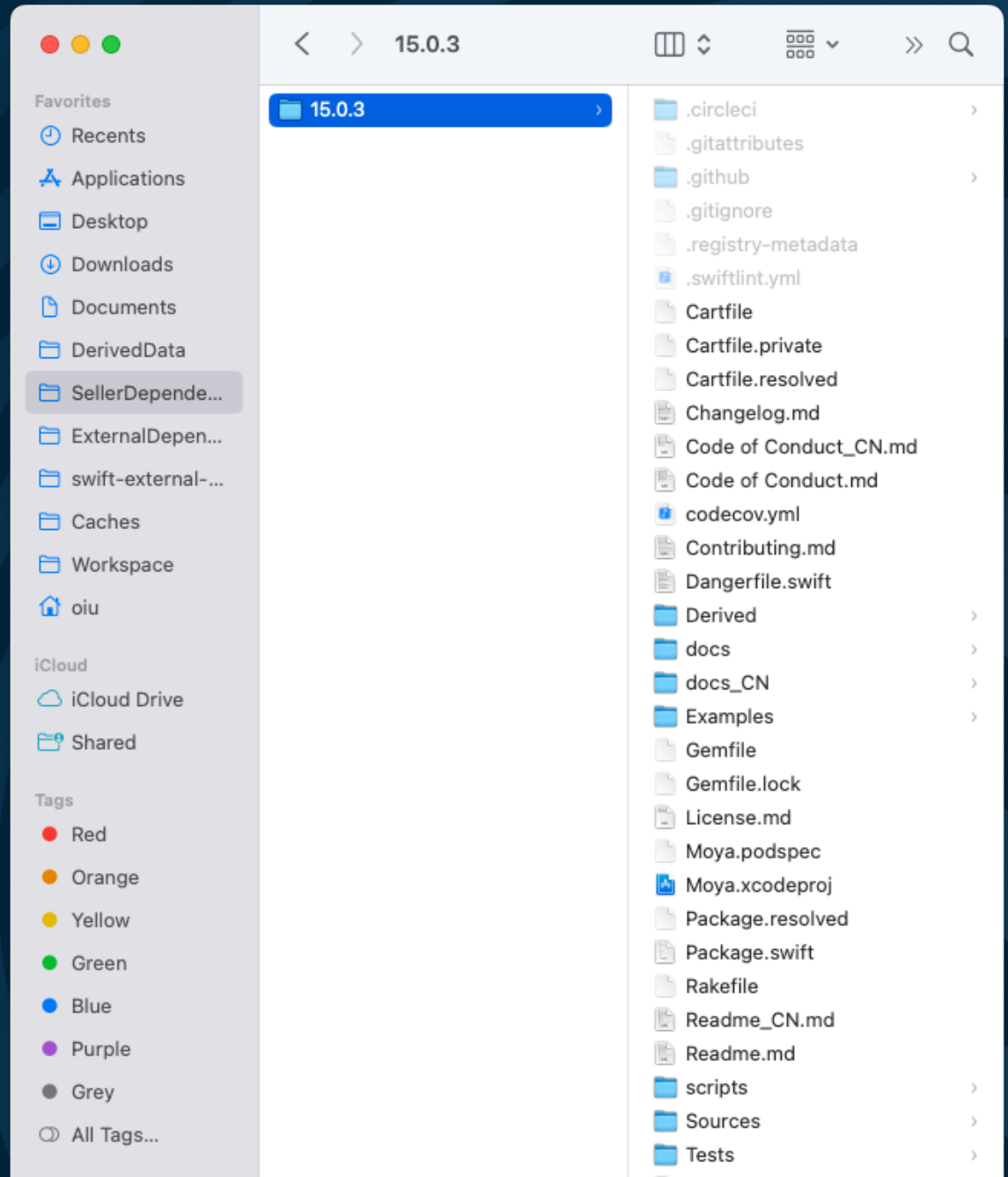
```jsonc
// registries.json
{
  "authentication" : {

  },
  "registries" : {
    "leet-scope" : {
      "supportsAvailability" : false,
      "url" : "sanya.org/"
    },
  },
  "version" : 1
}
```

ozontech

Как работает сервис

А как выглядит на тачке?

```swift
public struct RegistryClient: RepositoryProvider {

    // ...

    func _downloadSourceArchive( // <- строка 967
        registry: Registry,
        package: PackageIdentity.RegistryIdentity,
        version: Version,
        destinationPath: AbsolutePath,
        progressHandler: ((_ bytesReceived: Int64, _ totalBytes: Int64?) -> Void)?,
        timeout: DispatchTimeInterval?,
        fileSystem: FileSystem,
        observabilityScope: ObservabilityScope,
        callbackQueue: DispatchQueue,
        completion: @escaping (Result<Void, Error>) -> Void
    ) {
      // ... скачка, валидации чексумм и подписей
    } // <- строка 1200
}
```

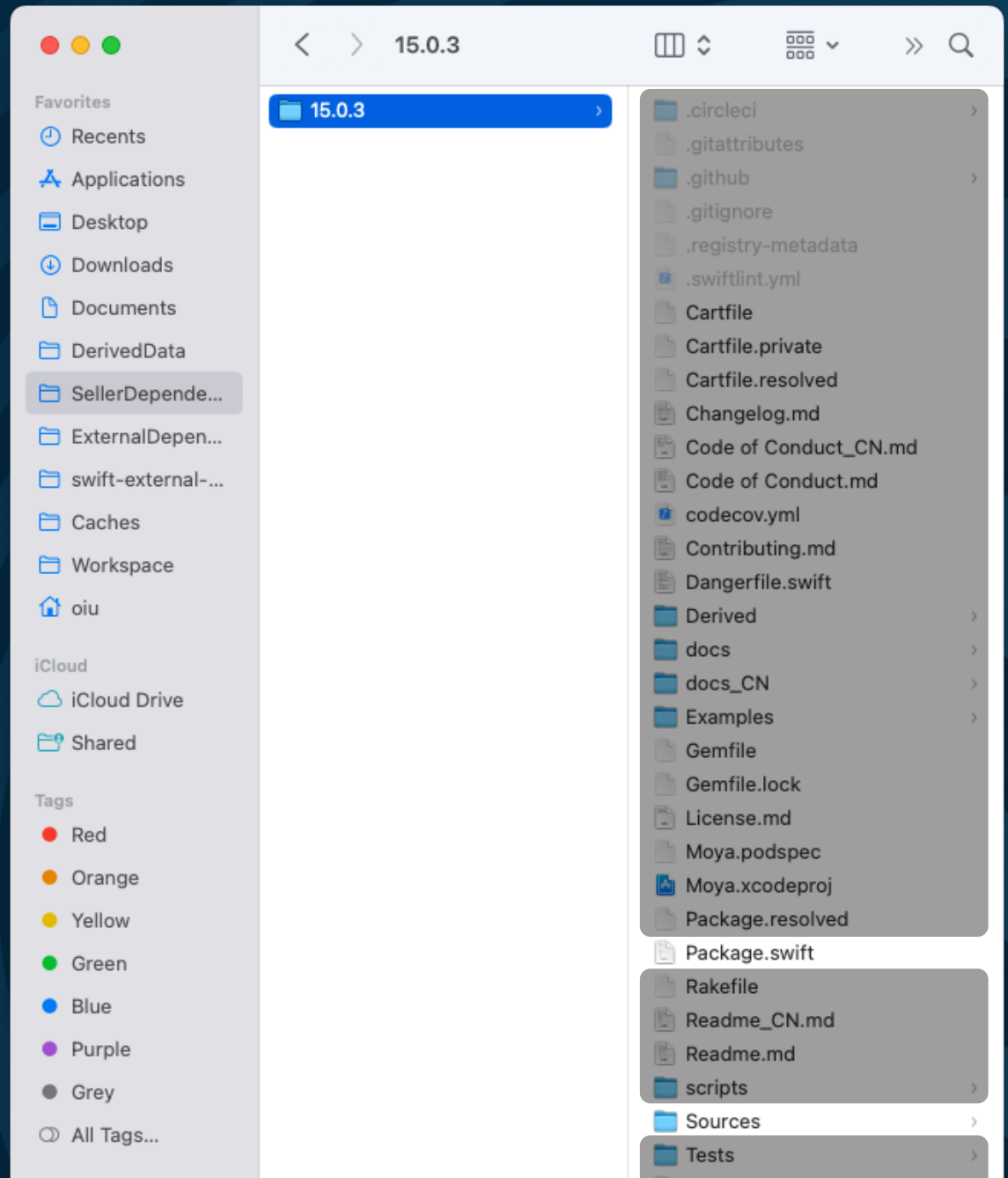


ozontech

Как работает сервис

А как выглядит на тачке?

```swift
public struct RegistryClient: RepositoryProvider {

    // ...

    func _downloadSourceArchive( // <- строка 967
        registry: Registry,
        package: PackageIdentity.RegistryIdentity,
        version: Version,
        destinationPath: AbsolutePath,
        progressHandler: ((_ bytesReceived: Int64, _ totalBytes: Int64?) -> Void)?,
        timeout: DispatchTimeInterval?,
        fileSystem: FileSystem,
        observabilityScope: ObservabilityScope,
        callbackQueue: DispatchQueue,
        completion: @escaping (Result<Void, Error>) -> Void
    ) {
      // ... скачка, валидации чексумм и подписей
    } // <- строка 1200
}
```



ozontech

03

В чём бенефиты

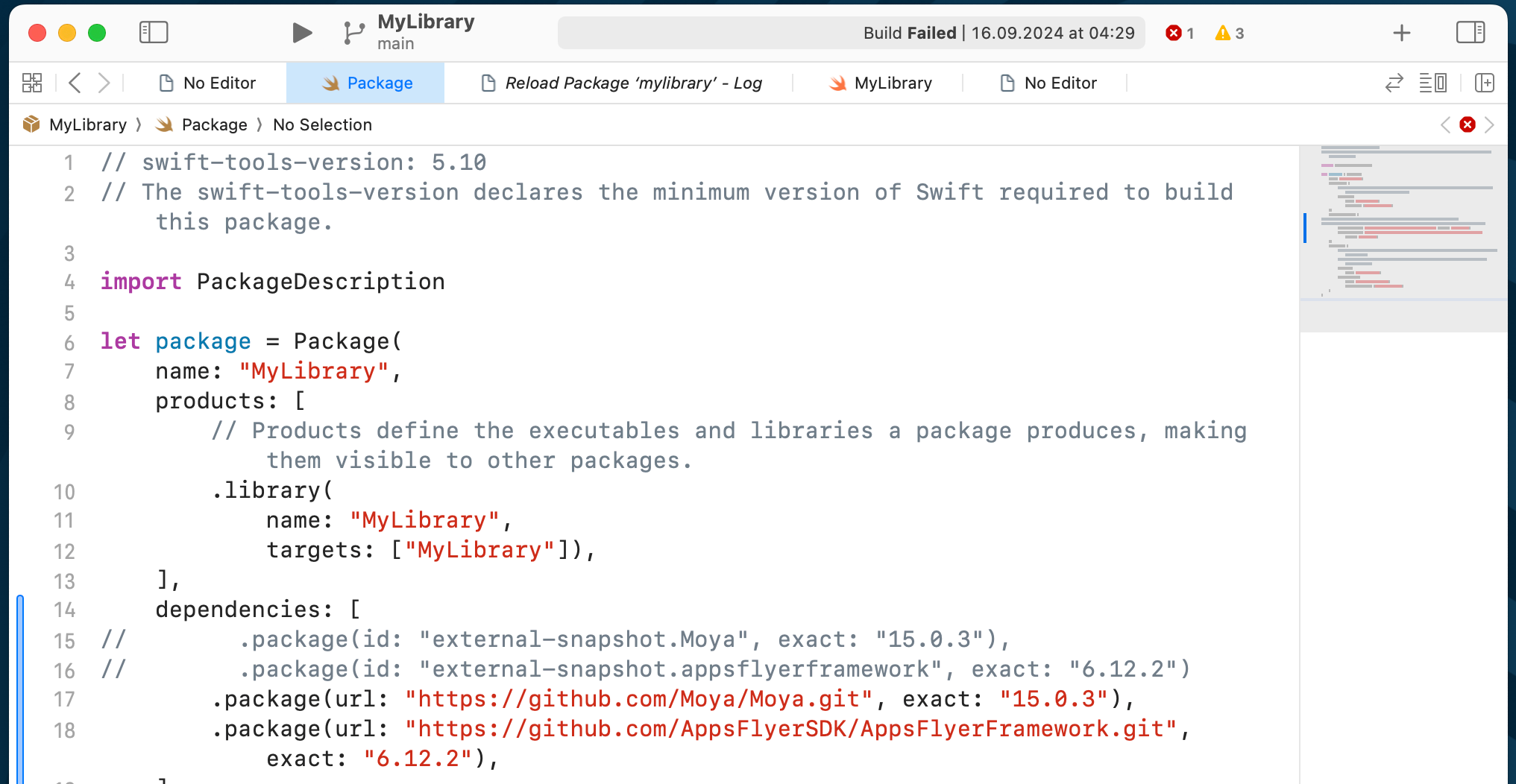
И где нюансы

Что получилось в итоге?

	SPM	SPR
Скорость	–	+
Эксплуатация и обслуживание	+	–

Почему скорость

Прямое сравнение



Почему скорость

Прямое сравнение

```
~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4 (6m 11.63s)
swift package purge-cache; \
swift package reset; \
swift package resolve

Fetched https://github.com/ReactiveCocoa/ReactiveSwift.git from cache
(22.43s)
Fetched https://github.com/ReactiveX/RxSwift.git from cache (22.84s)
Computing version for https://github.com/ReactiveX/RxSwift.git
Computed https://github.com/ReactiveX/RxSwift.git at 6.7.1 (0.53s)
Computing version for https://github.com/ReactiveCocoa/ReactiveSwift.g
it
Computed https://github.com/ReactiveCocoa/ReactiveSwift.git at 6.7.0 (
0.75s)
Computing version for https://github.com/Alamofire/Alamofire.git
Computed https://github.com/Alamofire/Alamofire.git at 5.9.1 (0.42s)
Creating working copy for https://github.com/AppsFlyerSDK/AppsFlyerFra
mework.git
Working copy of https://github.com/AppsFlyerSDK/AppsFlyerFramework.git
 resolved at 6.12.2
Creating working copy for https://github.com/ReactiveCocoa/ReactiveSwi
ft.git
Working copy of https://github.com/ReactiveCocoa/ReactiveSwift.git res
olved at 6.7.0
Creating working copy for https://github.com/Moya/Moya.git
Working copy of https://github.com/Moya/Moya.git resolved at 15.0.3
Creating working copy for https://github.com/Alamofire/Alamofire.git
Working copy of https://github.com/Alamofire/Alamofire.git resolved at
 5.9.1
Creating working copy for https://github.com/ReactiveX/RxSwift.git
Working copy of https://github.com/ReactiveX/RxSwift.git resolved at 6
.7.1
Fetching binary artifact https://github.com/AppsFlyerSDK/AppsFlyerFram
ework/releases/download/6.12.2/AppsFlyerLib.xcframework.zip from cache
```

```
~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4 (0.016s)
ls


~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4 (9.212s)
swift package purge-cache; \
swift package reset; \
swift package resolve

Fetching external-snapshot.rxswift
warning: 'external-snapshot.rxswift': external-snapshot.rxswift versio
n 6.2.0 source archive from
                                    is not signed
Fetched external-snapshot.rxswift from cache (1.42s)
Fetching external-snapshot.Moya
warning: 'external-snapshot.Moya': external-snapshot.Moya version 15.0
.3 source archive from
                                is not signed
Fetched external-snapshot.Moya from cache (0.23s)
Fetching external-snapshot.appsflyerframework
warning: 'external-snapshot.appsflyerframework': external-snapshot.app
sflyerframework version 6.12.2 source archive from
                                                          is not signed
Fetched external-snapshot.appsflyerframework from cache (0.07s)
Fetching external-snapshot.alamofire
warning: 'external-snapshot.alamofire': external-snapshot.alamofire ve
rsion 5.6.2 source archive from
                                    is not signed
Fetched external-snapshot.alamofire from cache (0.89s)
Fetching external-snapshot.reactiveswift
warning: 'external-snapshot.reactiveswift': external-snapshot.reactive
swift version 6.6.1 source archive from
                                              is not signed
```

Почему скорость

Прямое сравнение

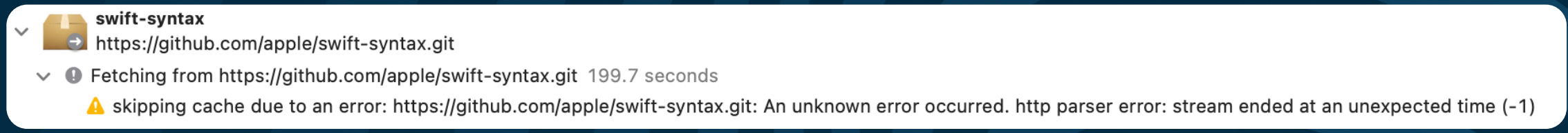
```
~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±3 (28.65s)
swift package resolve
Computed https://github.com/Moya/Moya.git at 15.0.3 (0.02s)
Fetching https://github.com/ReactiveCocoa/ReactiveSwift.git from cache
Fetching https://github.com/ReactiveX/RxSwift.git from cache
Fetching https://github.com/Alamofire/Alamofire.git from cache
Fetched https://github.com/Alamofire/Alamofire.git from cache (1.54s)
Fetched https://github.com/ReactiveCocoa/ReactiveSwift.git from cache
(1.88s)
Fetched https://github.com/ReactiveX/RxSwift.git from cache (2.21s)
Computing version for https://github.com/ReactiveX/RxSwift.git
Computed https://github.com/ReactiveX/RxSwift.git at 6.7.1 (0.02s)
Computing version for https://github.com/ReactiveCocoa/ReactiveSwift.g
it
Computed https://github.com/ReactiveCocoa/ReactiveSwift.git at 6.7.0 (
0.02s)
Computing version for https://github.com/Alamofire/Alamofire.git
Computed https://github.com/Alamofire/Alamofire.git at 5.9.1 (0.03s)
Creating working copy for https://github.com/Moya/Moya.git
Working copy of https://github.com/Moya/Moya.git resolved at 15.0.3
Creating working copy for https://github.com/ReactiveX/RxSwift.git
Working copy of https://github.com/ReactiveX/RxSwift.git resolved at 6
.7.1
Creating working copy for https://github.com/AppsFlyerSDK/AppsFlyerFra
mework.git
Working copy of https://github.com/AppsFlyerSDK/AppsFlyerFramework.git
 resolved at 6.12.2
Creating working copy for https://github.com/Alamofire/Alamofire.git
Working copy of https://github.com/Alamofire/Alamofire.git resolved at
 5.9.1
Creating working copy for https://github.com/ReactiveCocoa/ReactiveSwi
ft.git
```

```
~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4 (0.016s)
ls


~/Workspace/Registry/Registry/Features/MyLibrary git:(main)±4 (9.212s)
swift package purge-cache; \
swift package reset; \
swift package resolve
Fetching external-snapshot.rxswift
warning: 'external-snapshot.rxswift': external-snapshot.rxswift versio
n 6.2.0 source archive from
                                    is not signed
Fetched external-snapshot.rxswift from cache (1.42s)
Fetching external-snapshot.Moya
warning: 'external-snapshot.Moya': external-snapshot.Moya version 15.0
.3 source archive from
                                is not signed
Fetched external-snapshot.Moya from cache (0.23s)
Fetching external-snapshot.appsflyerframework
warning: 'external-snapshot.appsflyerframework': external-snapshot.app
sflyerframework version 6.12.2 source archive from
                                                            is not signed
Fetched external-snapshot.appsflyerframework from cache (0.07s)
Fetching external-snapshot.alamofire
warning: 'external-snapshot.alamofire': external-snapshot.alamofire ve
rsion 5.6.2 source archive from
                                        is not signed
Fetched external-snapshot.alamofire from cache (0.89s)
Fetching external-snapshot.reactiveswift
warning: 'external-snapshot.reactiveswift': external-snapshot.reactive
swift version 6.6.1 source archive from
                                                is not signed
```

Почему скорость

Что у SPM?

- Чекауты могут быть долгими на пустой кэш
- Одни чекауты задерживают другие (простройка графа)
- Сначала загружаем пакет и разбираем его манифест, а уже потом переходим к его зависимостям
- Кэш работает нестабильно (и его много)



Почему скорость

Что у SPR?

- Есть отдельная ручка на запрос всех версий
- Есть отдельная ручка на архив с самим пакетом
- Кэши сильно легче и не протухают (просчёт через md5)

)))

Почему эксплуатация
У SPM

- Типичная жизнь с гитом
  - Иногда постреливающие креды
  - Внешний репозиторий может быть недоступен
  - Тег, на который мы завязались, может быть снесён
- В остальном — все привыкли

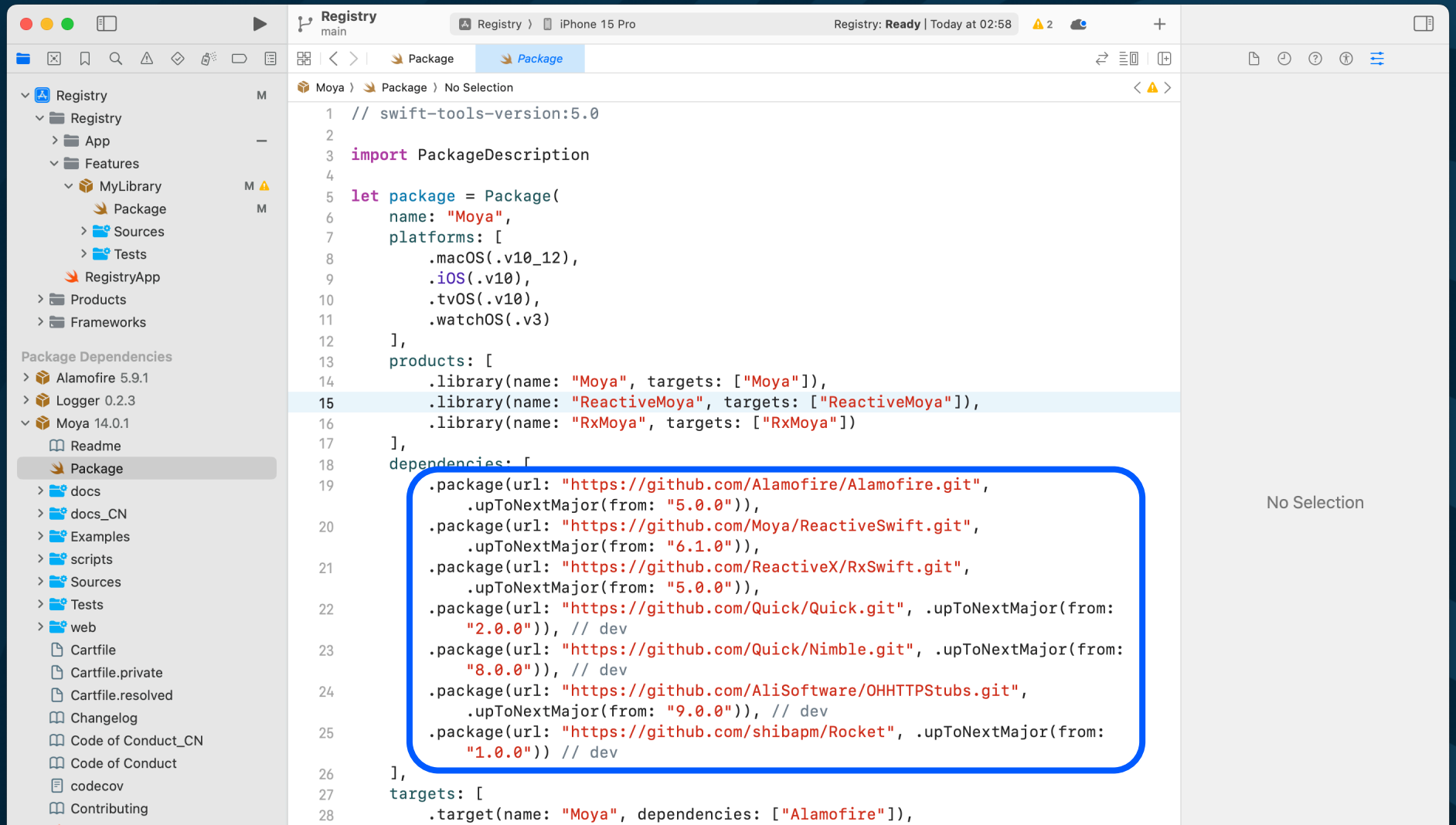
Почему эксплуатация

У SPR

- Нужно проставлять конфиги `swift package-registry set`
- Скорее всего — проставлять зеркала с гитовых реп
- … есть нюанс)

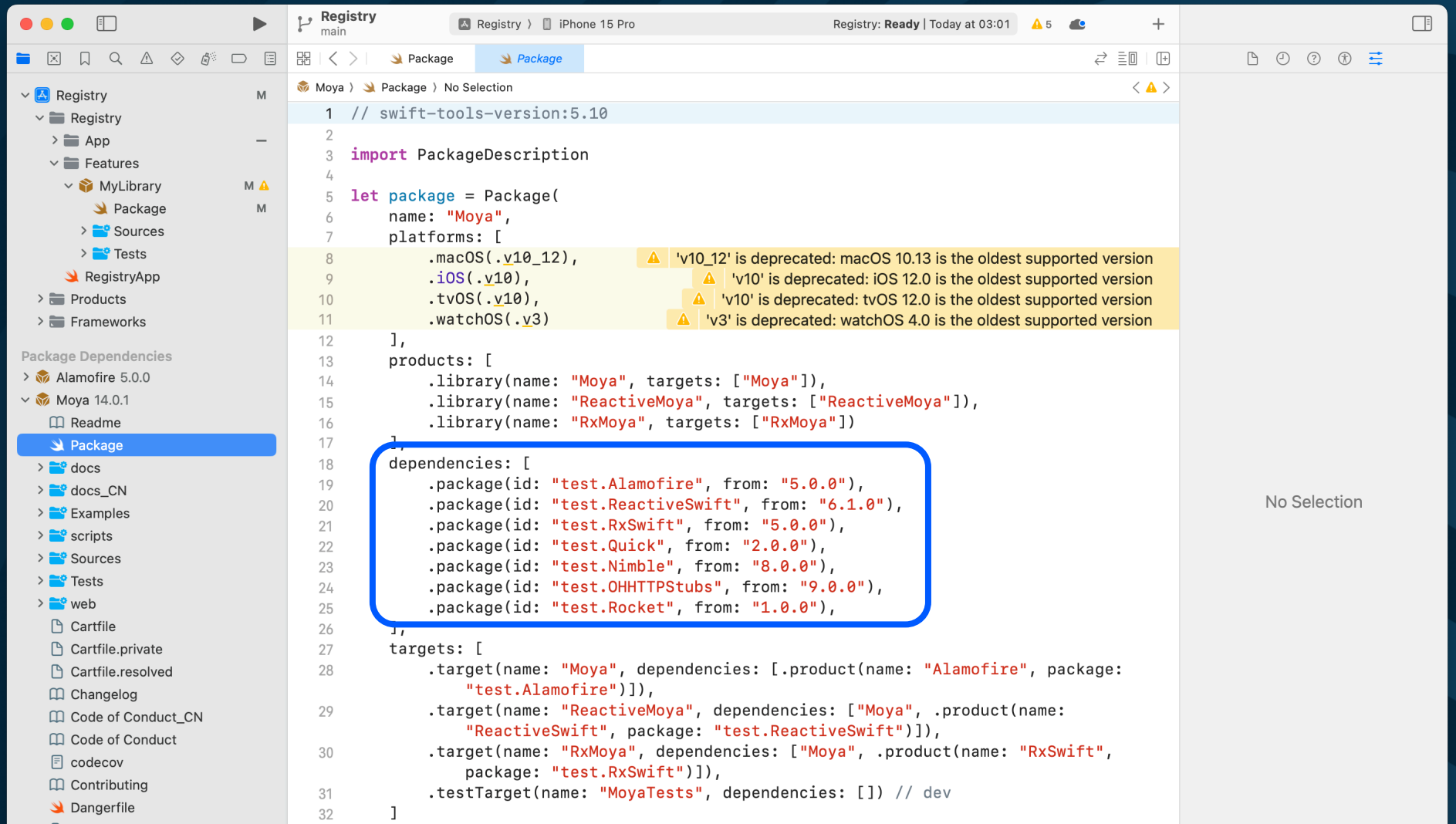
Почему эксплуатация

А где брать registry зависимости?



Почему эксплуатация

А где брать registry зависимости?



ozontech

Почему эксплуатация

Версионирование транзитивных зависимостей

```
.package(
    url: "https://github.com/Moya/Moya.git",
    exact: "15.0.3"
),
// ... depends on
.package(
    url: "https://github.com/Alamofire/Alamofire.git",
    exact: "5.6.2"
),
```

```
.package(
    url: "https://github.com/Moya/Moya.git",
    exact: "15.0.3"
),
// ... depends on
.package(
    url: "https://github.com/Alamofire/Alamofire.git",
    branch: "update/xcode-13.3"
),
```

Почему эксплуатация

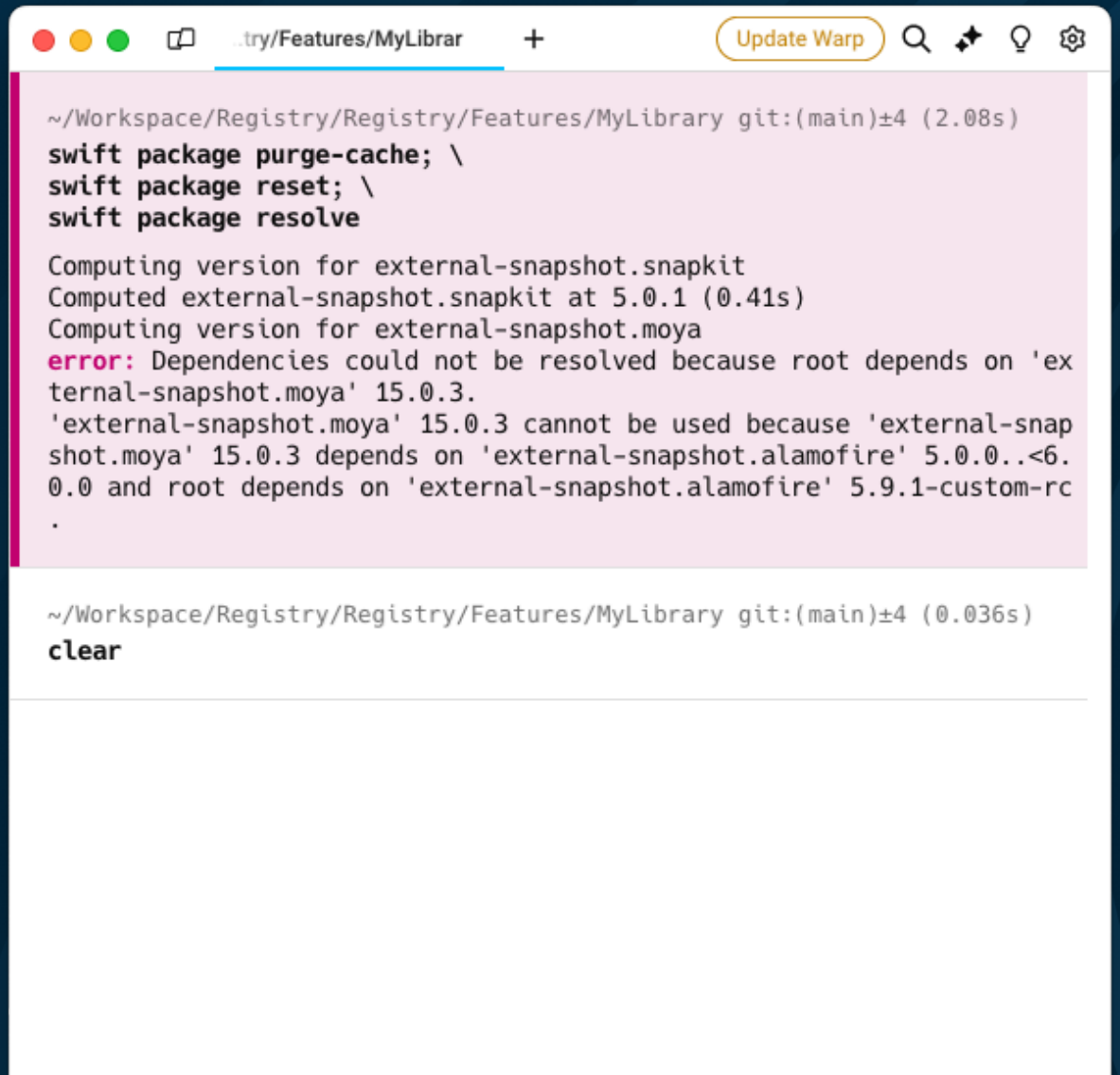
Версионирование транзитивных зависимостей

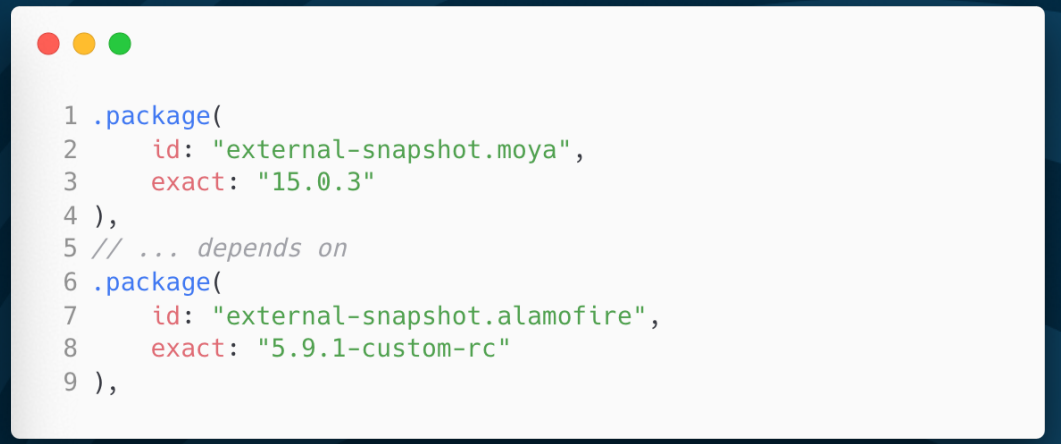
```
.package(
    id: "external-snapshot.moya",
    exact: "15.0.3"
),
// ... depends on
.package(
    id: "external-snapshot.alamofire",
    exact: "5.6.2"
),
```

```
.package(
    id: "external-snapshot.moya",
    exact: "15.0.3"
),
// ... depends on
.package(
    id: "external-snapshot.alamofire",
    exact: "5.9.1-custom-rc"
),
```

Почему эксплуатация

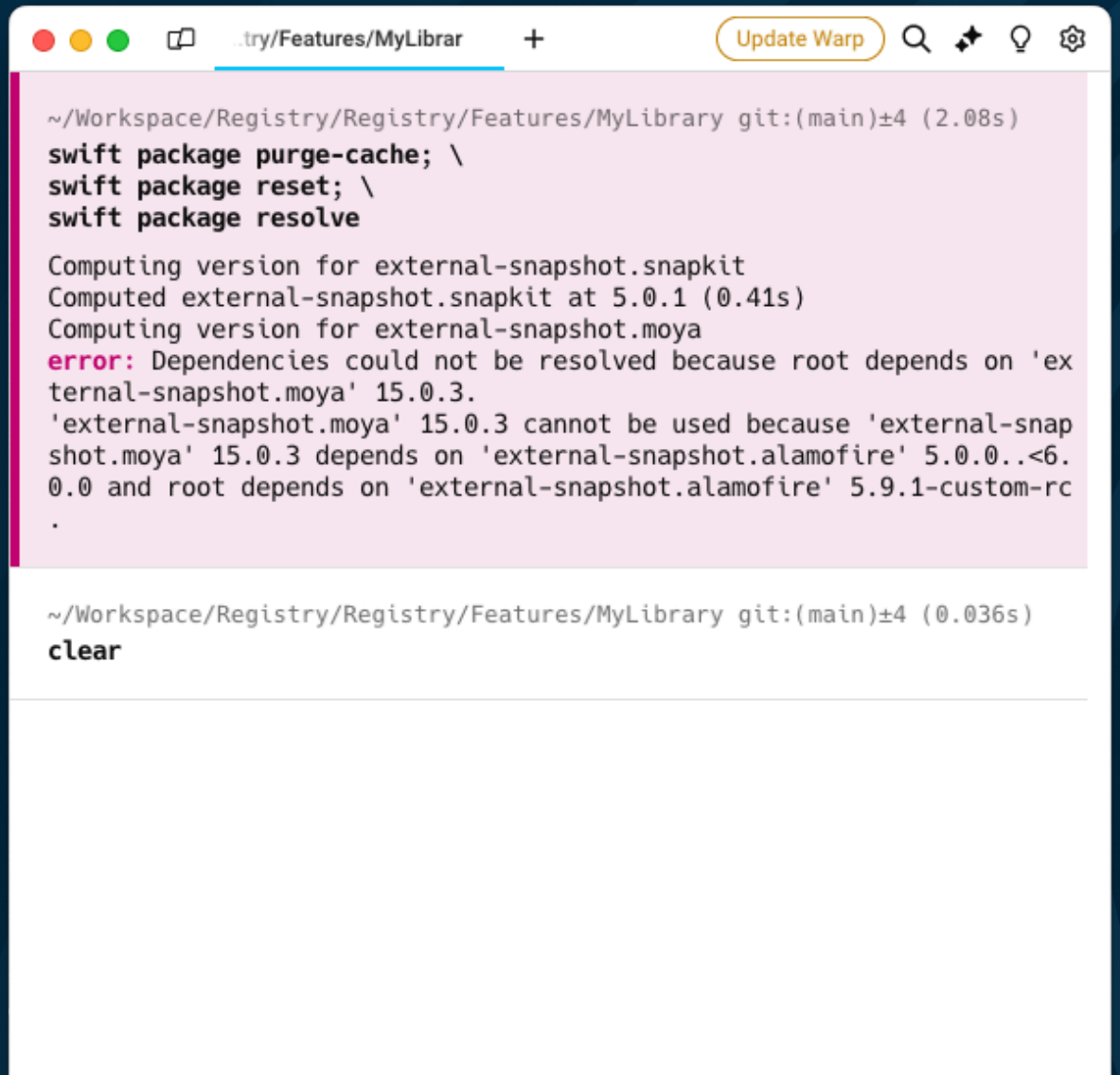
Версионирование транзитивных зависимостей

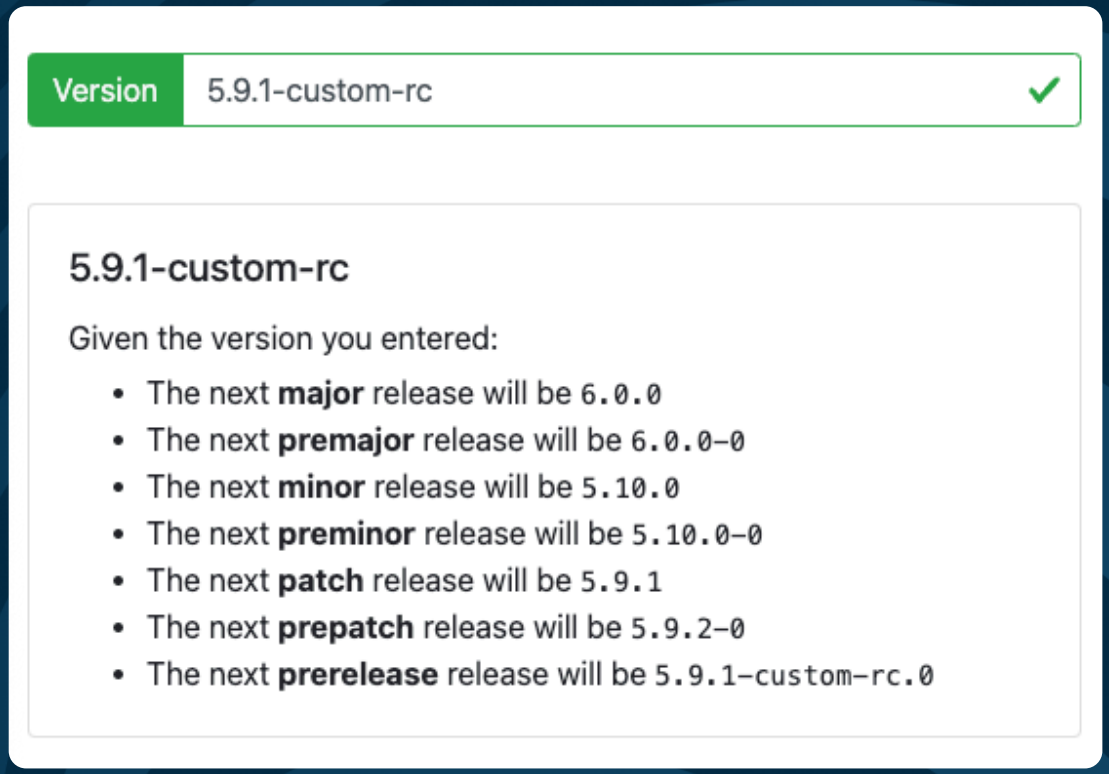




Почему эксплуатация

Версионирование транзитивных зависимостей





Почему обслуживание

SPM + SPR

- За исключением стандартных требований к работе с SPM (контроль подключений и источников), нас всё устраивает, пока сторонние Git-репозитории, с которых мы забираем нужные зависимости, работают стабильно

- По registry:
  - Для использования registry нужны свои серверы со своей имплементацией спеки, с которой сможет «общаться Xcode», а также мониторинг конфигураций (проставление скоупов) — это требует денежных средств
  - Плюсы: больше контроля. Минусы: больше контроля

Почему обслуживание

Нюансы перевода проектов с SPM -> SPR

Registry — не самая стабильная штука при работе с обычными готовыми пакетами

- Постреливает кэш из-за identity пакетов, которые раньше грузились с git

Если вы релизились только в git, а потом резко начали релизиться только в registry — **у вас проблемы**

- При переводе проектов и обновлении версий часто приходится опираться на зеркала, иначе будем ловить дубликаты identity
- Проект может неправильно разрешить зависимости: если нужной версии нет в registry, SPM автоматически подтянет пакет с соответствующей identity из git. Однако этот пакет может не подходить по ABI. Резолв пройдёт успешно, но сборка будет падать

Как можно использовать registry?

Безопасность и стабильность

1. Так как в registry пакеты приходится выгружать — их в любом случае надо как-то обрабатывать
2. Раз уж пакеты приходится трогать (автоматикой или вручную), их можно проверять на уязвимости в безопасности
3. Перепроверка и валидация фингерпринтов и подписей
4. Выгружая все пакеты на локальный ресурс, можно в конце концов полностью уйти от использования внешних сервисов (например, GitHub)
5. Можно контролировать доступ к любым библиотекам, если все клиенты смотрят только на registry (dependency whitelist)

Как можно использовать registry?

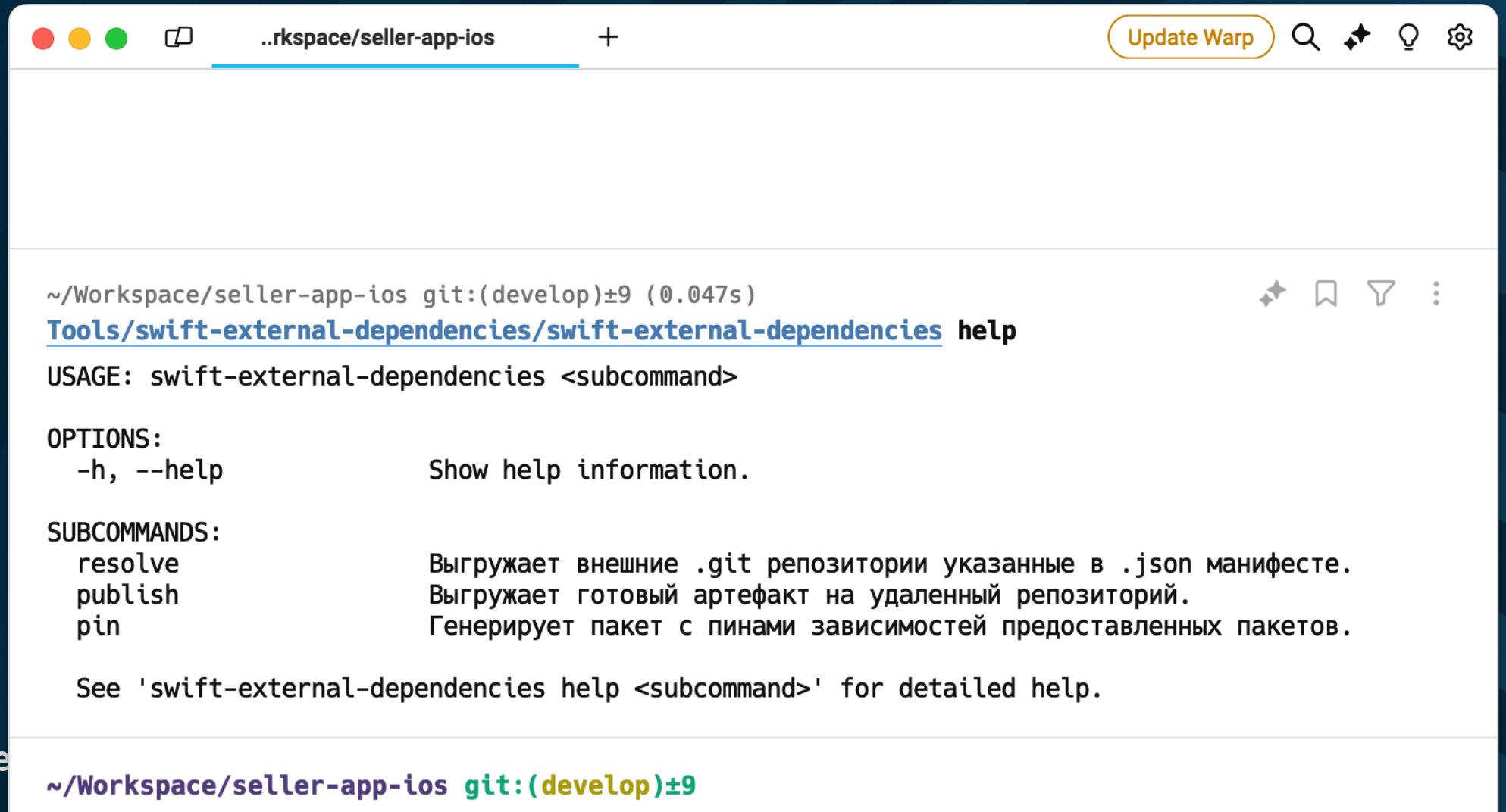
swift-external-dependencies

swift-external-dependencies и воркфлоу разработчика

- В репозитории лежит .json файл, который является апрувнутым белым листом внешних зависимостей (spm пакетов)
- Разработчик коммитит в репозиторий изменение списка зависимостей и получает апрувы от безопасников
- После апрува и мержа, внешний репозиторий и его транзитивные зависимости выгружаются в нашу registry

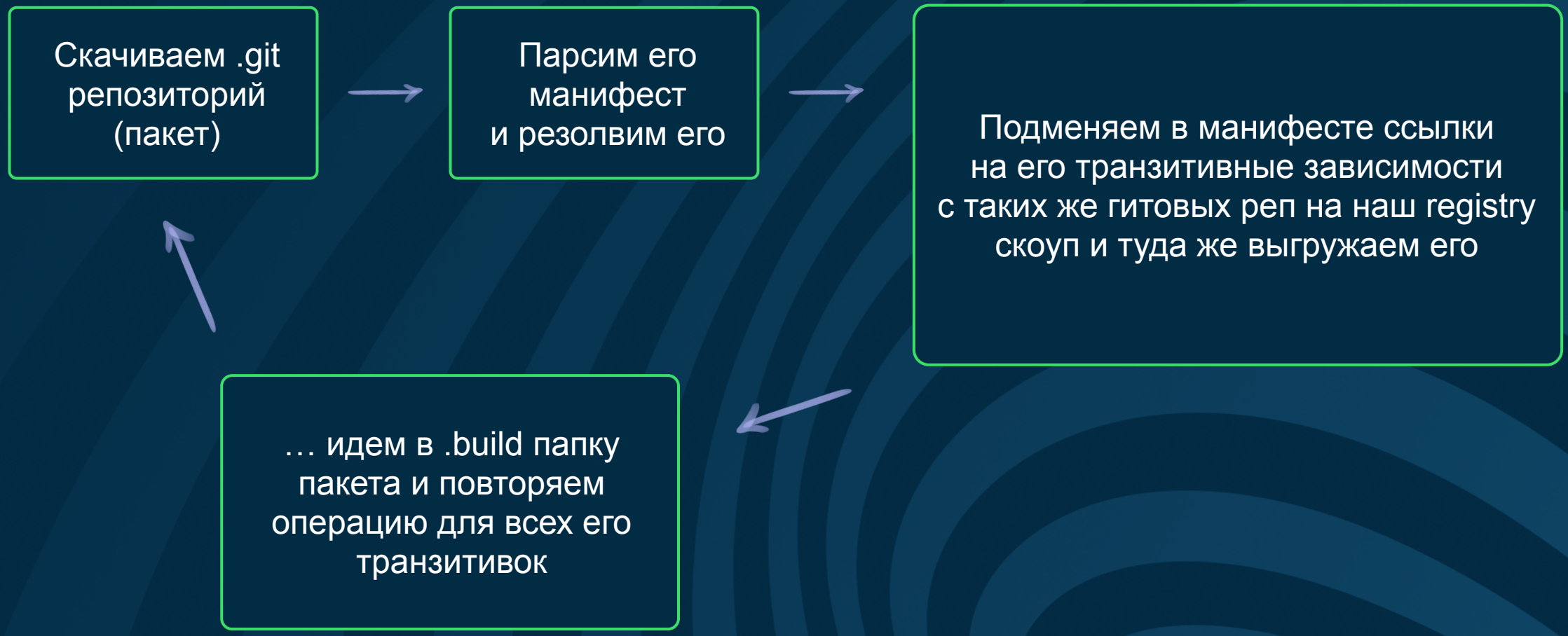
Как можно использовать registry?

swift-external-dependencies



Как можно использовать registry?

swift-external-dependencies

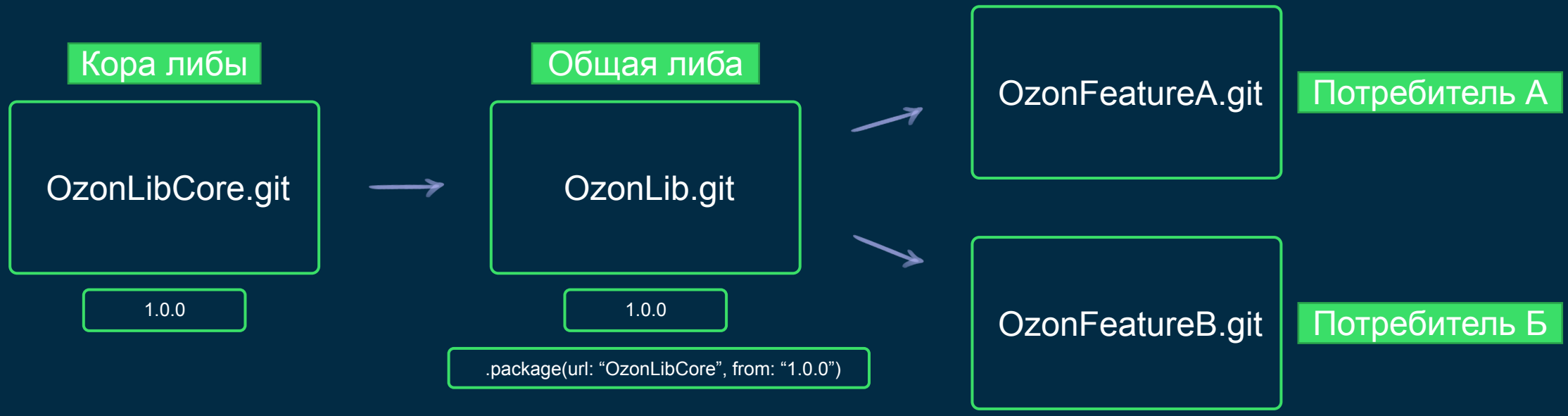


Как можно использовать registry?

Монорепа для внутренних библиотек и инструментов

Мы построили монорепу на основе registry — хотите так же?

- Зачем нужна?

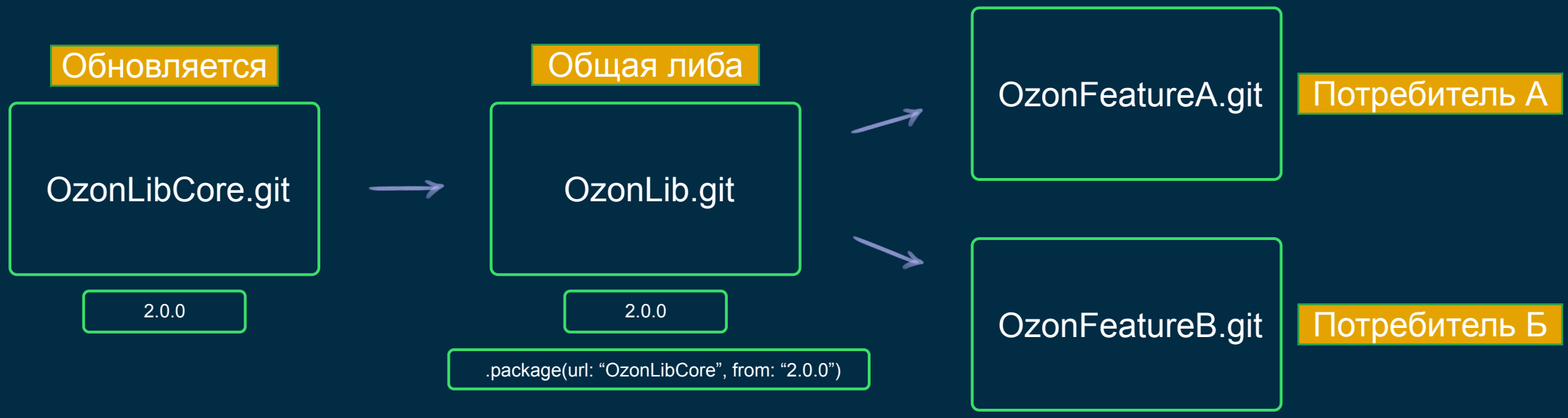


Как можно использовать registry?

Монорепа для внутренних библиотек и инструментов

Мы построили монорепу на основе registry — хотите так же?

- Зачем нужна?

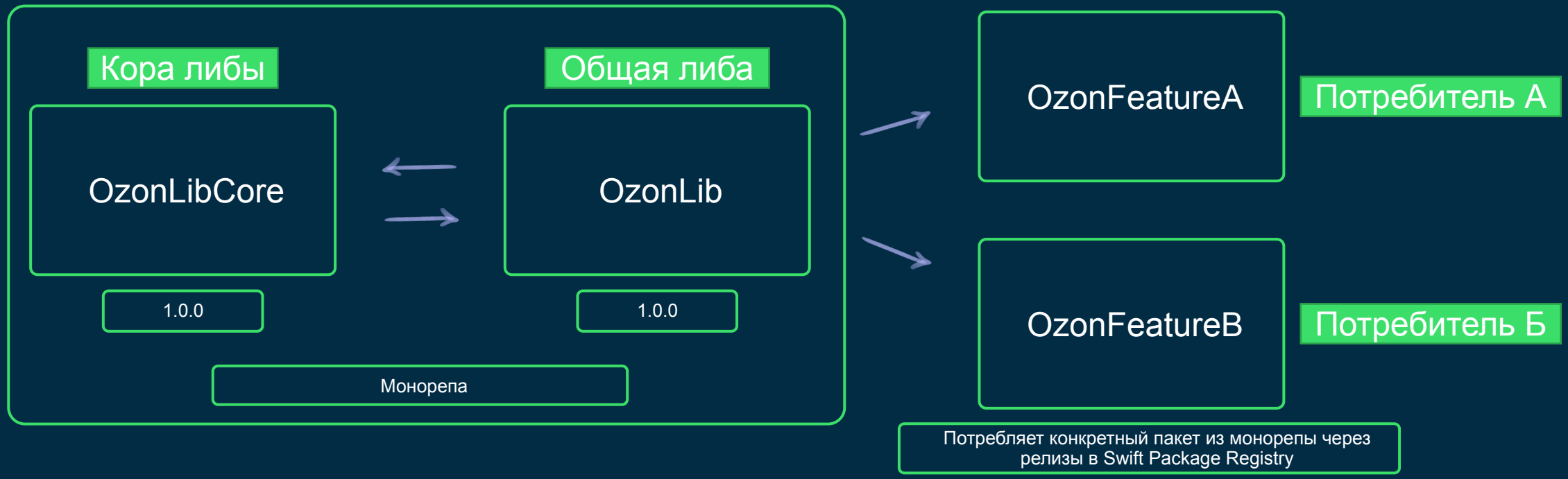


Как можно использовать registry?

Монорепа для внутренних библиотек и инструментов

Мы построили монорепу на основе registry — хотите так же?

- Зачем нужна?



Как можно использовать registry?

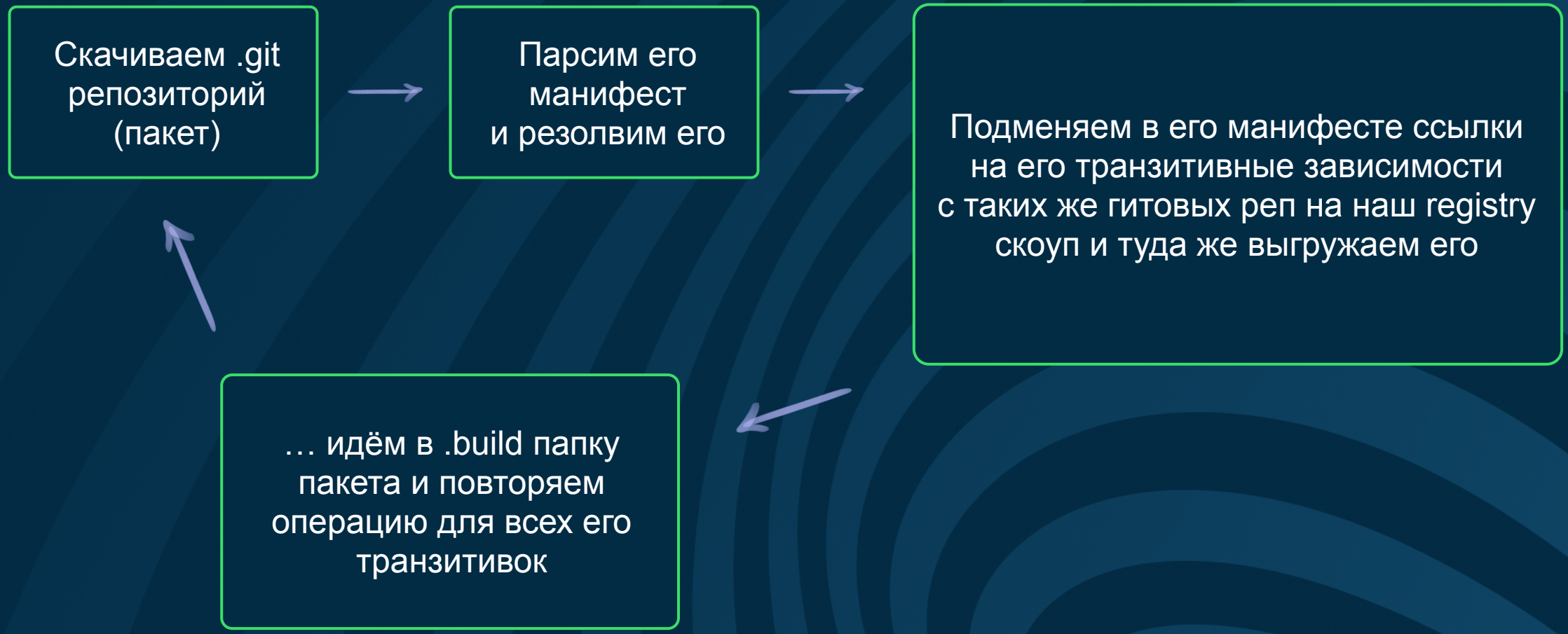
Монорепа для внутренних библиотек и инструментов

Мы построили монорепу на основе registry — хотите так же?

- Монорепа с разбойными релизами
- Работаем со всеми исходниками, а на релизы выгружаем отдельный архив конкретного пакета из монорепы (сильно меньше dependency hell при релизе нескольких пакетов)
- Без монорепы невозможно (либо с костылями)
- Попробуйте как-нибудь положить пакет в пакет

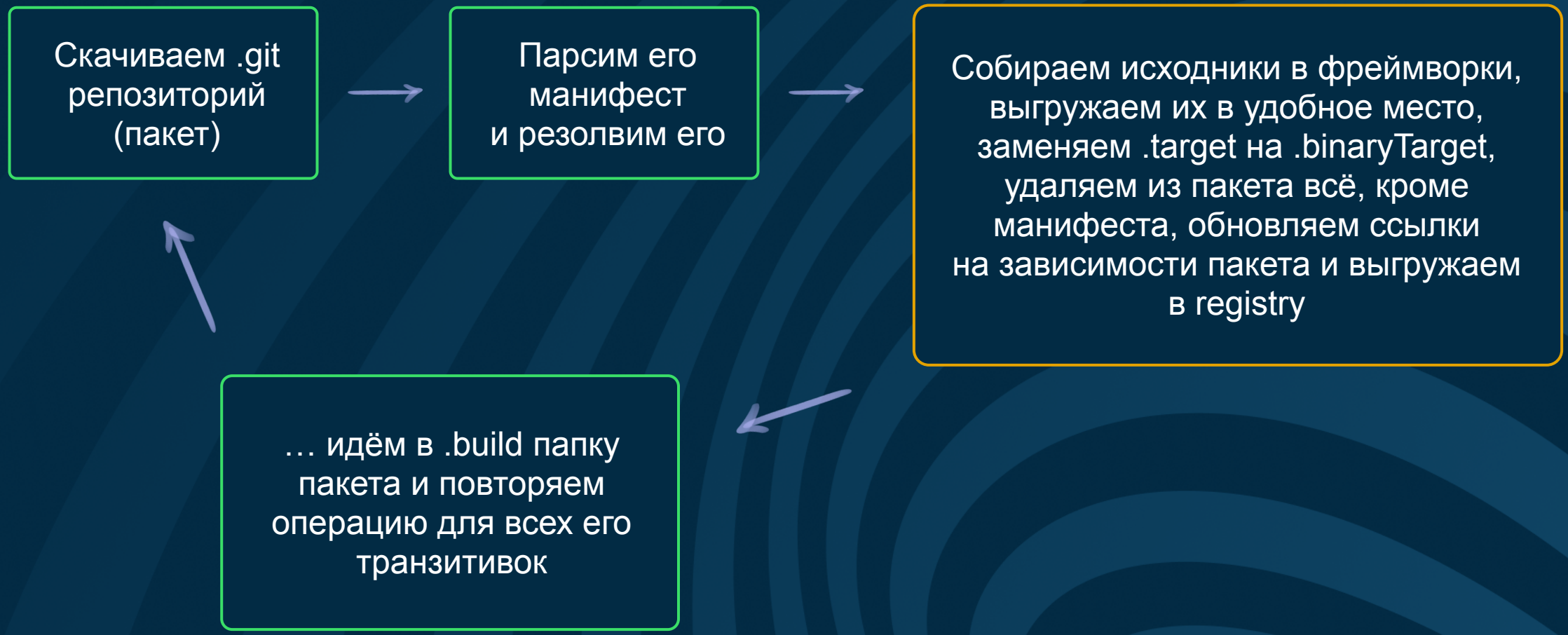
Как можно использовать registry?

Система кэширования



Как можно использовать registry?

Система кеширования



Как можно использовать registry?

Скорость

Система кэширования на пакетах — а зачем?

- Такого просто нет (привет Tuist)
- Никакой обвязки на стороне клиента (проекта) ((Привет Tuist))
- Простая подмена сурсов на бинари / наоборот через зеркала на нужные скоупы
- Можно и на гите, но с registry сильно проще

04

Заключение

Как оно?

Заключение

Общее впечатление

- SPM уже, так или иначе, стал стандартом разработки
  - Registry позволяет решить часть проблем SPM
- Registry позволяет решить проблемы не только самого SPM, но и реальные нужды вашего конкретного проекта
- Registry быстрее

ozon tech

Заключение

Субъективное впечатление

- Переход на registry в текущем её состоянии должен происходить **ТОЧНО** не за цифры

- Если переходите (особенно не SPM -> SPR, а например Pods -> SPM) — готовьтесь к страданиям

ozon tech

Полезные материалы

Локальная имплементация Swift Package Registry

JFrog (Artifactory) Documentation

Swift Registry Service Open-api

ozontech

OZON tech

Спасибо за внимание, вопросы?

Евгений Рыжов
Senior iOS Developer, Ozon