Swift Registry

Dependencies Outside of GitHub

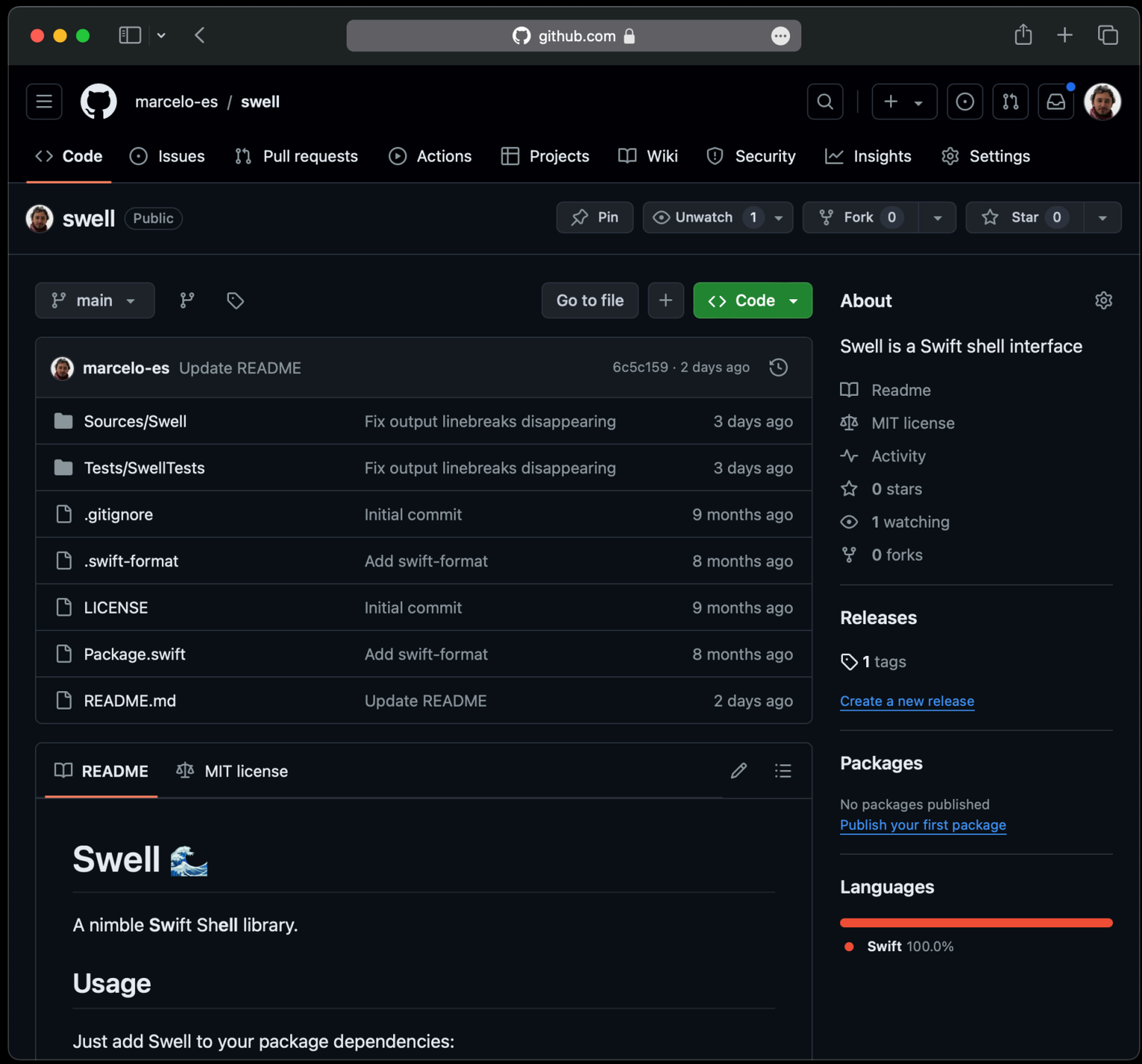
Marcelo Esperidiao, Engineer @ANZx

@marcelo@mastodon.au | linkedin.com/in/marceloes

Swift Figlet 🎨

Swift Figlet 🎨 →

Swell 🌊

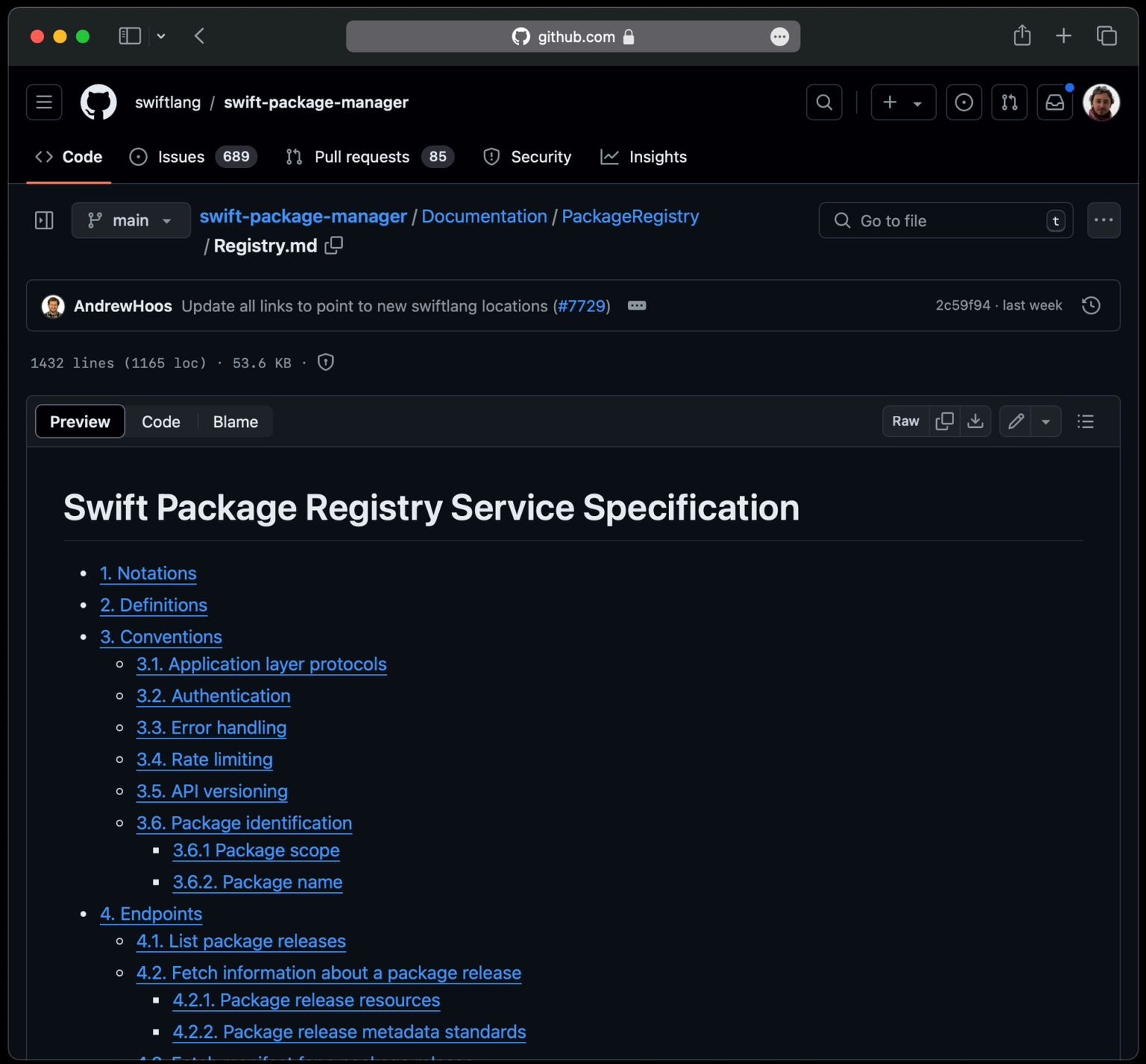

github.com
marcelo-es / swell
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
swell
Public
Pin
Unwatch 1
Fork 0
Star 0
main
Go to file
Code
marcelo-es Update README
6c5c159 · 2 days ago
Sources/Swell
Fix output linebreaks disappearing
3 days ago
Tests/SwellTests
Fix output linebreaks disappearing
3 days ago
.gitignore
Initial commit
9 months ago
.swift-format
Add swift-format
8 months ago
LICENSE
Initial commit
9 months ago
Package.swift
Add swift-format
8 months ago
README.md
Update README
2 days ago
README
MIT license
Swell 🌊
A nimble Swift Shell library.
Usage
Just add Swell to your package dependencies:
About
Swell is a Swift shell interface
Readme
MIT license
Activity
0 stars
1 watching
0 forks
Releases
1 tags
Create a new release
Packages
No packages published
Publish your first package
Languages
Swift 100.0%

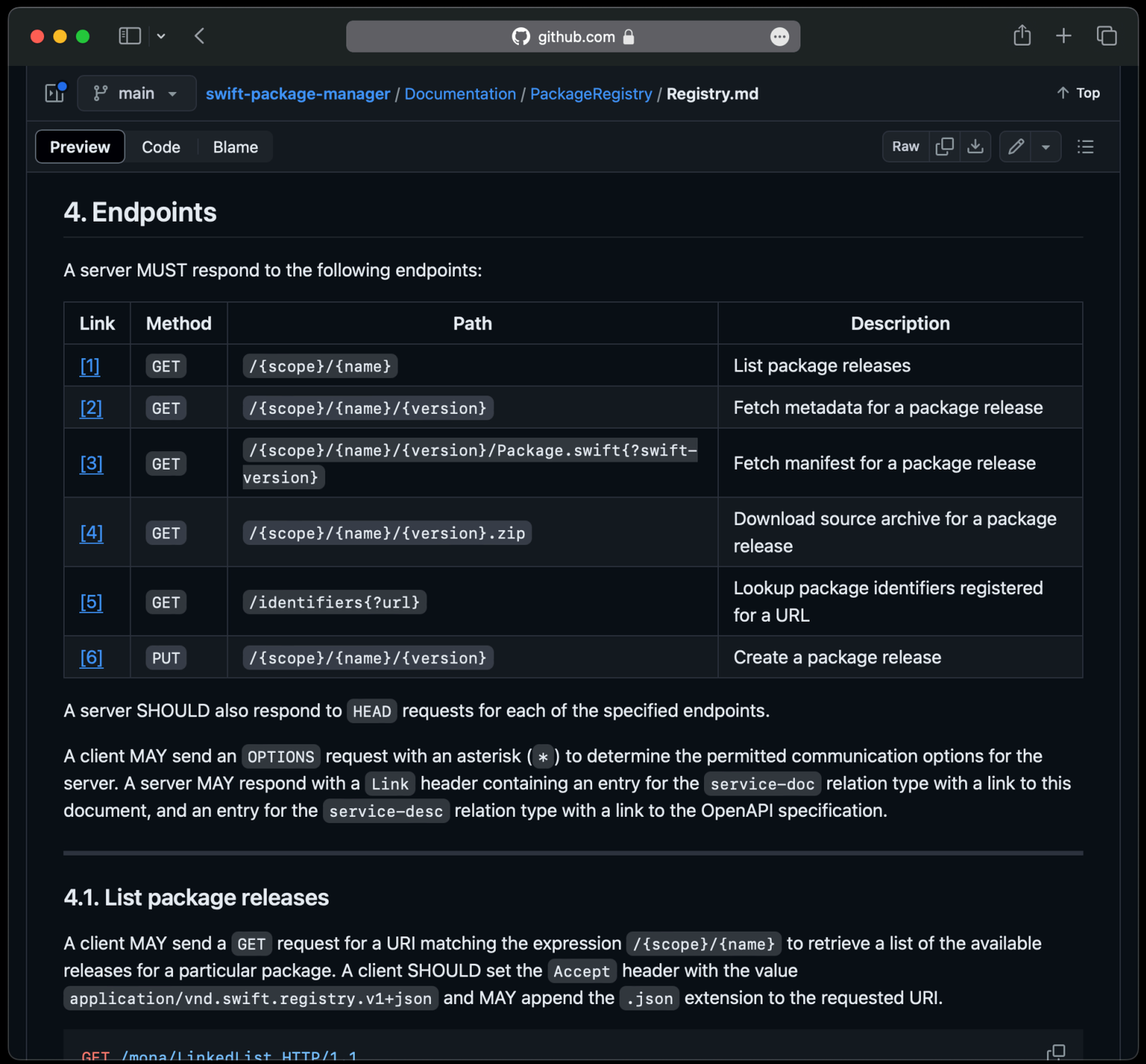
What if Swell 🌊 can't be on GitHub?

Swift Registry

(anvil animation goes here)

What sort of  is this?


github.com
swiftlang / swift-package-manager
Code
Issues 689
Pull requests 85
Security
Insights
main
swift-package-manager / Documentation / PackageRegistry / Registry.md
Go to file
t
AndrewHoos Update all links to point to new swiftlang locations (#7729)
2c59f94 · last week
1432 lines (1165 loc) · 53.6 KB ·
Preview
Code
Blame
Raw
Swift Package Registry Service Specification
1. Notations
2. Definitions
3. Conventions
3.1. Application layer protocols
3.2. Authentication
3.3. Error handling
3.4. Rate limiting
3.5. API versioning
3.6. Package identification
3.6.1 Package scope
3.6.2. Package name
4. Endpoints
4.1. List package releases
4.2. Fetch information about a package release
4.2.1. Package release resources
4.2.2. Package release metadata standards



4. Endpoints

A server MUST respond to the following endpoints:

Link	Method	Path	Description
[1]	`GET`	`/{scope}/{name}`	List package releases
[2]	`GET`	`/{scope}/{name}/{version}`	Fetch metadata for a package release
[3]	`GET`	`/{scope}/{name}/{version}/Package.swift{?swift-version}`	Fetch manifest for a package release
[4]	`GET`	`/{scope}/{name}/{version}.zip`	Download source archive for a package release
[5]	`GET`	`/identifiers{?url}`	Lookup package identifiers registered for a URL
[6]	`PUT`	`/{scope}/{name}/{version}`	Create a package release

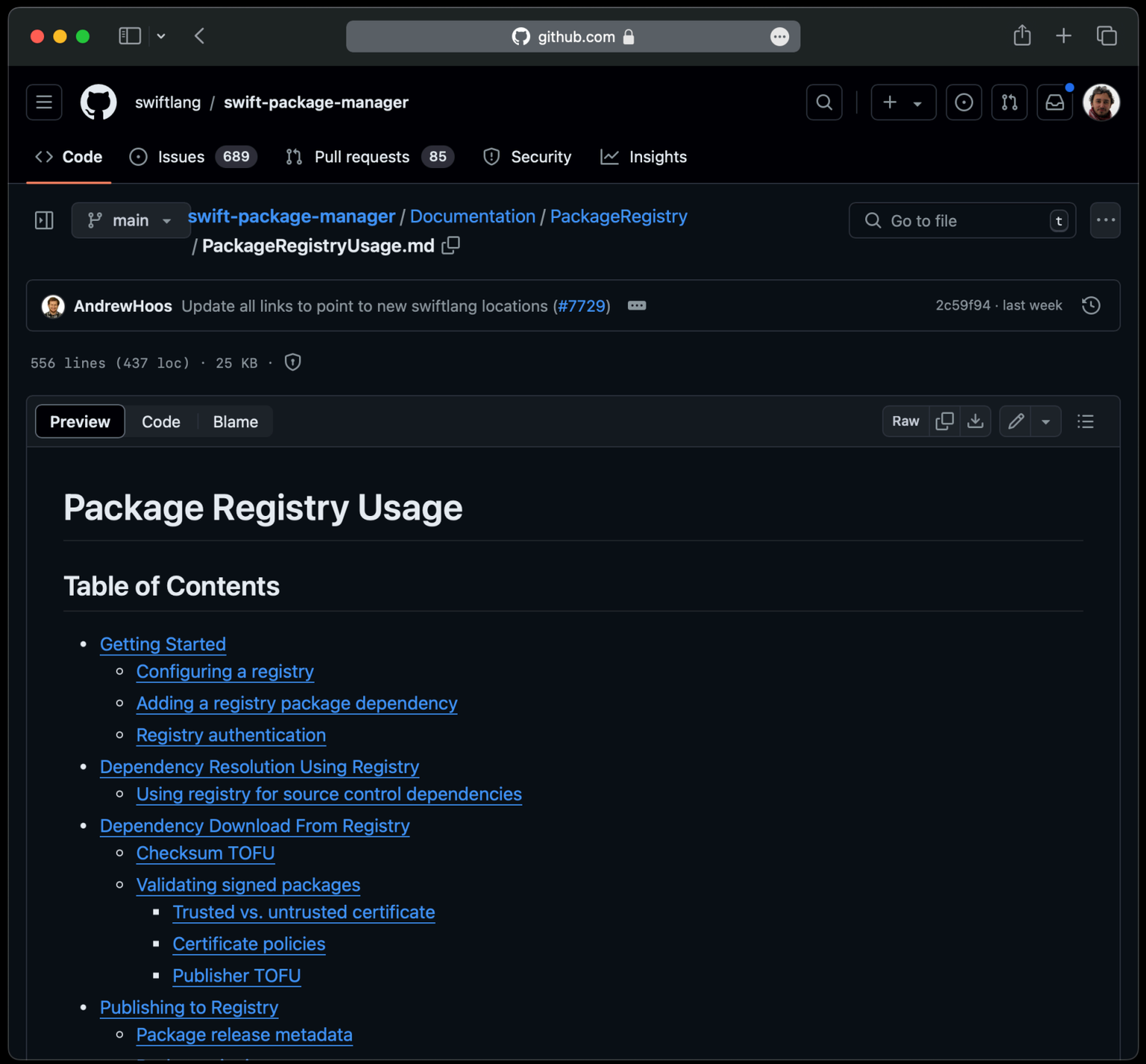
A server SHOULD also respond to `HEAD` requests for each of the specified endpoints.

A client MAY send an `OPTIONS` request with an asterisk (`*`) to determine the permitted communication options for the server. A server MAY respond with a `Link` header containing an entry for the `service-doc` relation type with a link to this document, and an entry for the `service-desc` relation type with a link to the OpenAPI specification.

4.1. List package releases

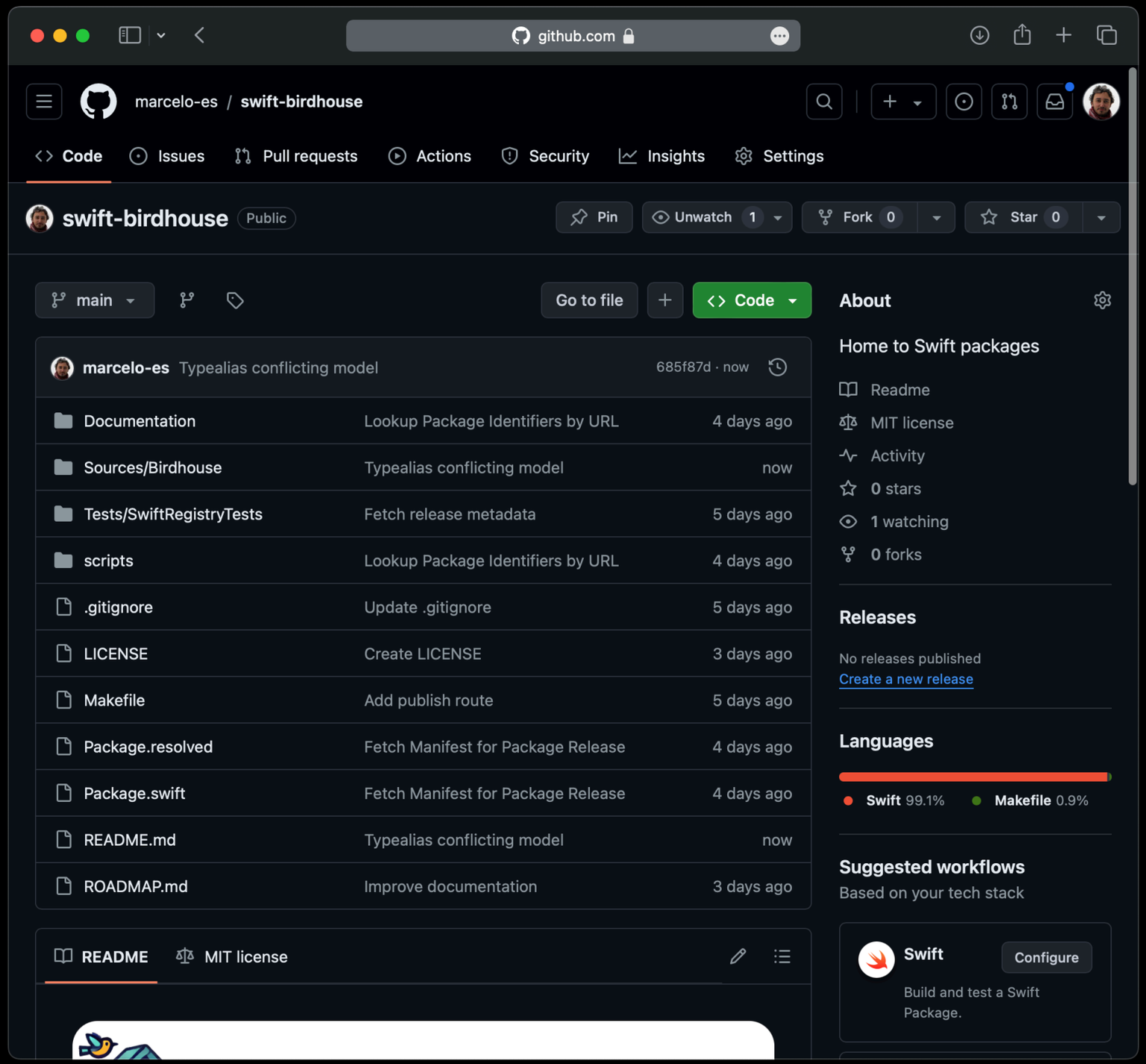
A client MAY send a `GET` request for a URI matching the expression `/{scope}/{name}` to retrieve a list of the available releases for a particular package. A client SHOULD set the `Accept` header with the value `application/vnd.swift.registry.v1+json` and MAY append the `.json` extension to the requested URI.

```
GET /mona/LinkedList HTTP/1.1
```


github.com
swiftlang / swift-package-manager
Code
Issues 689
Pull requests 85
Security
Insights
main
swift-package-manager / Documentation / PackageRegistry
/ PackageRegistryUsage.md
Go to file
t
AndrewHoos Update all links to point to new swiftlang locations (#7729)
2c59f94 · last week
556 lines (437 loc) · 25 KB ·
Preview
Code
Blame
Raw
Package Registry Usage
Table of Contents
Getting Started
Configuring a registry
Adding a registry package dependency
Registry authentication
Dependency Resolution Using Registry
Using registry for source control dependencies
Dependency Download From Registry
Checksum TOFU
Validating signed packages
Trusted vs. untrusted certificate
Certificate policies
Publisher TOFU
Publishing to Registry
Package release metadata

Swift Registry is the spec, so where did Swell 🌊 go?

(logo by ChatGPT)


github.com
marcelo-es / swift-birdhouse
Code
Issues
Pull requests
Actions
Security
Insights
Settings
swift-birdhouse
Public
Pin
Unwatch 1
Fork 0
Star 0
main
Go to file
Code
marcelo-es Typealias conflicting model
685f87d · now
Documentation Lookup Package Identifiers by URL 4 days ago
Sources/Birdhouse Typealias conflicting model now
Tests/SwiftRegistryTests Fetch release metadata 5 days ago
scripts Lookup Package Identifiers by URL 4 days ago
.gitignore Update .gitignore 5 days ago
LICENSE Create LICENSE 3 days ago
Makefile Add publish route 5 days ago
Package.resolved Fetch Manifest for Package Release 4 days ago
Package.swift Fetch Manifest for Package Release 4 days ago
README.md Typealias conflicting model now
ROADMAP.md Improve documentation 3 days ago
README
MIT license
About
Home to Swift packages
Readme
MIT license
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Languages
Swift 99.1%
Makefile 0.9%
Suggested workflows
Based on your tech stack
Swift
Configure
Build and test a Swift Package.

Swift Birdhouse

- Pure to the specs
- Built with Hummingbird 2 (`2.0.0-rc.1`)
- Has as few dependencies as we can get away with
- Blazing fast and light on runtime
- Not ready for production
  - yet!

Show me 🪵!

Scopes?

```swift
let package = Package(
    name: "swift-gremlins",
    dependencies: [
        .package(id: "byte-me.pixels", from: "2.1.0"),
        .package(id: "nerdvana.Glitch", from: "0.2.0"),
    ],
    targets: [
        .executableTarget(
            name: "swift-gremlins",
            dependencies: [
                .product(name: "Pixels", package: "byte-me.pixels"),
                .product(name: "Glitch", package: "nerdvana.Glitch"),
            ]
        ),
    ]
)
```

Package.swift

```
{
  "authentication" : {

  },
  "registries" : {
    "byte-me" : {
      "url" : "https://byte-me-solutions.com"
    },
    "nerdvana" : {
      "url" : "https://nerdvana.systems"
    }
  },
  "version" : 1
}
```

.swiftpm/configuration/registries.json

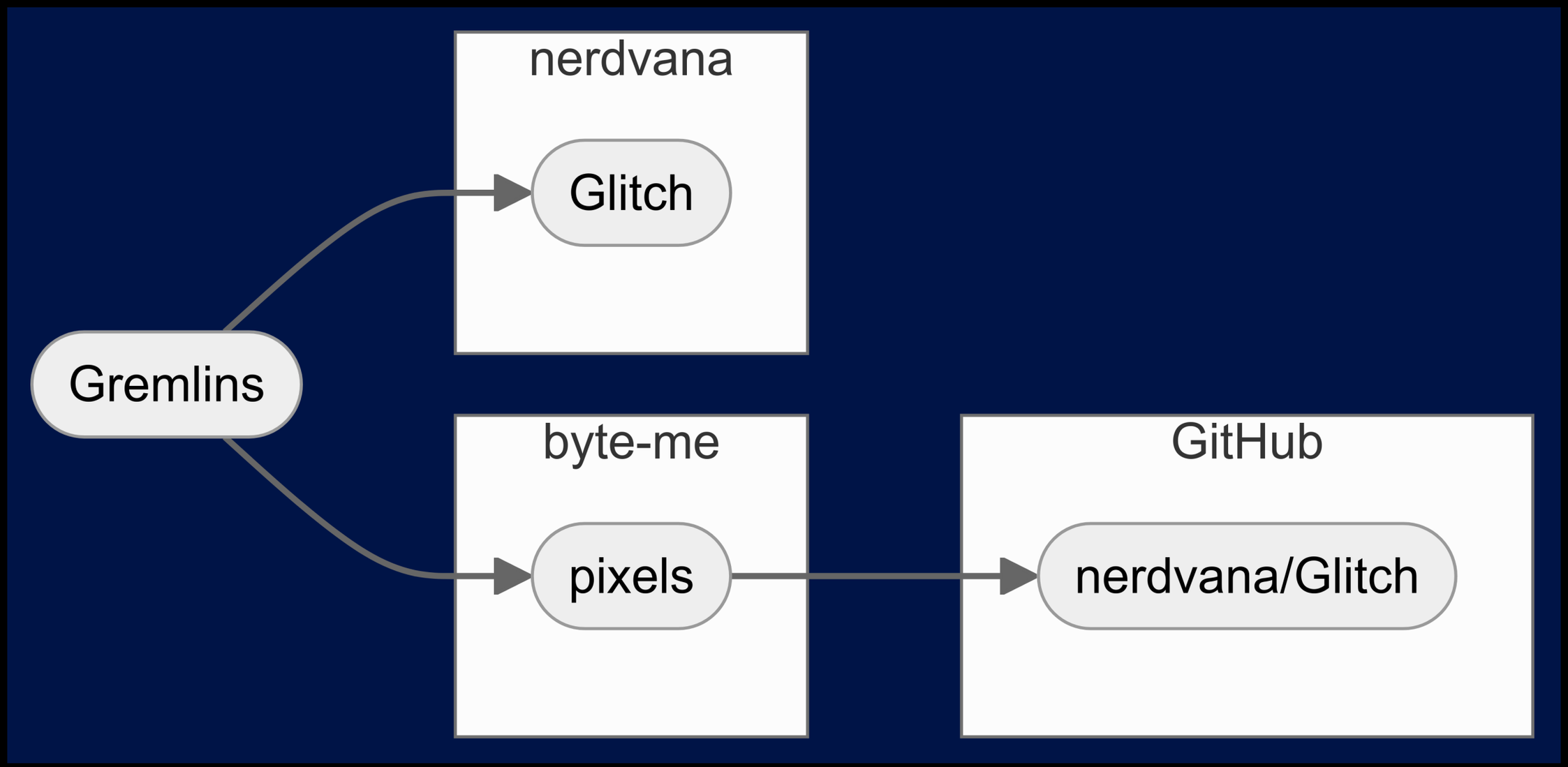
```
{
  "authentication" : {

  },
  "registries" : {
    "[default]" : {
      "url" : "https://byte-me-solutions.com"
    }
  },
  "version" : 1
}
```

.swiftpm/configuration/registries.json

What if dependencies conflict ⚔️?

```
let package = Package(
    name: "swift-gremlins",
    dependencies: [
        .package(
            id: "byte-me.pixels",
            from: "2.1.0"
        ),
        .package(
            id: "nerdvana.Glitch",
            from: "0.2.0"
        ),
    ],
    targets: [
        .executableTarget(
            name: "swift-gremlins",
            dependencies: [
                .product(
                    name: "Pixels",
                    package: "byte-me.pixels"
                ),
                .product(
                    name: "Glitch",
                    package: "nerdvana.Glitch"
                ),
            ]
        )
    ]
}
```

```
let package = Package(
    name: "pixels",
    dependencies: [
        .package(
            url: "https://github.com/nerdvana/Glitch",
            from: "0.2.0"
        ),
    ],
    targets: [
        .executableTarget(
            name: "Pixels",
            dependencies: [
                .product(
                    name: "Glitch",
                    package: "Glitch"
                ),
            ]
        ),
    ]
}
```


nerdvana
Glitch
Gremlins
byte-me
pixels
GitHub
nerdvana/Glitch

```
$ swift build --help

  --disable-scm-to-registry-transformation (default)

  --use-registry-identity-for-scm

  --replace-scm-with-registry
```

Thanks!

Questions?