*Swift Meetup Munich #35*

# Secure Your API Keys

## Harnessing Swift on Server

# About me

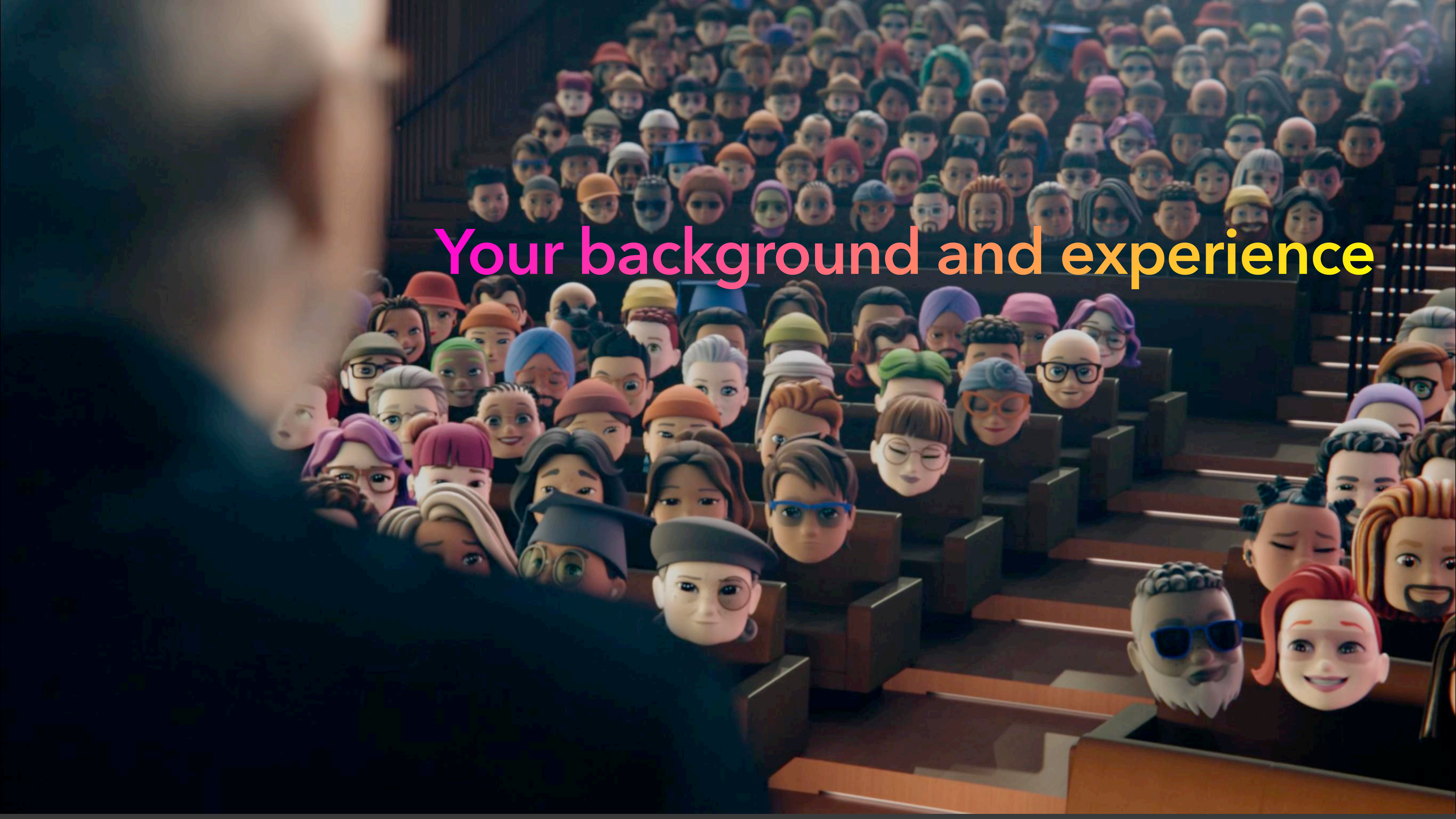
Michael Freiwald

📱 iOS Developer

👨‍💻 IT-Consultant @ Netlight

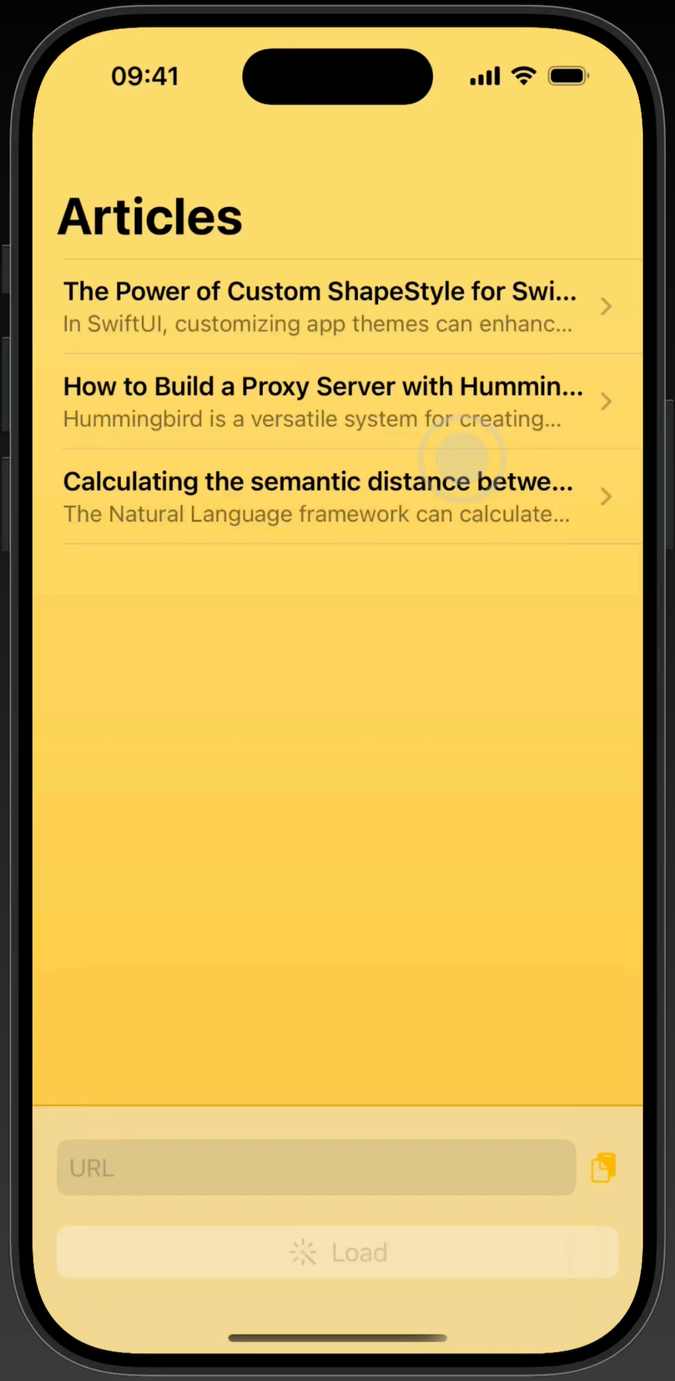
 Love writing Code with Swift

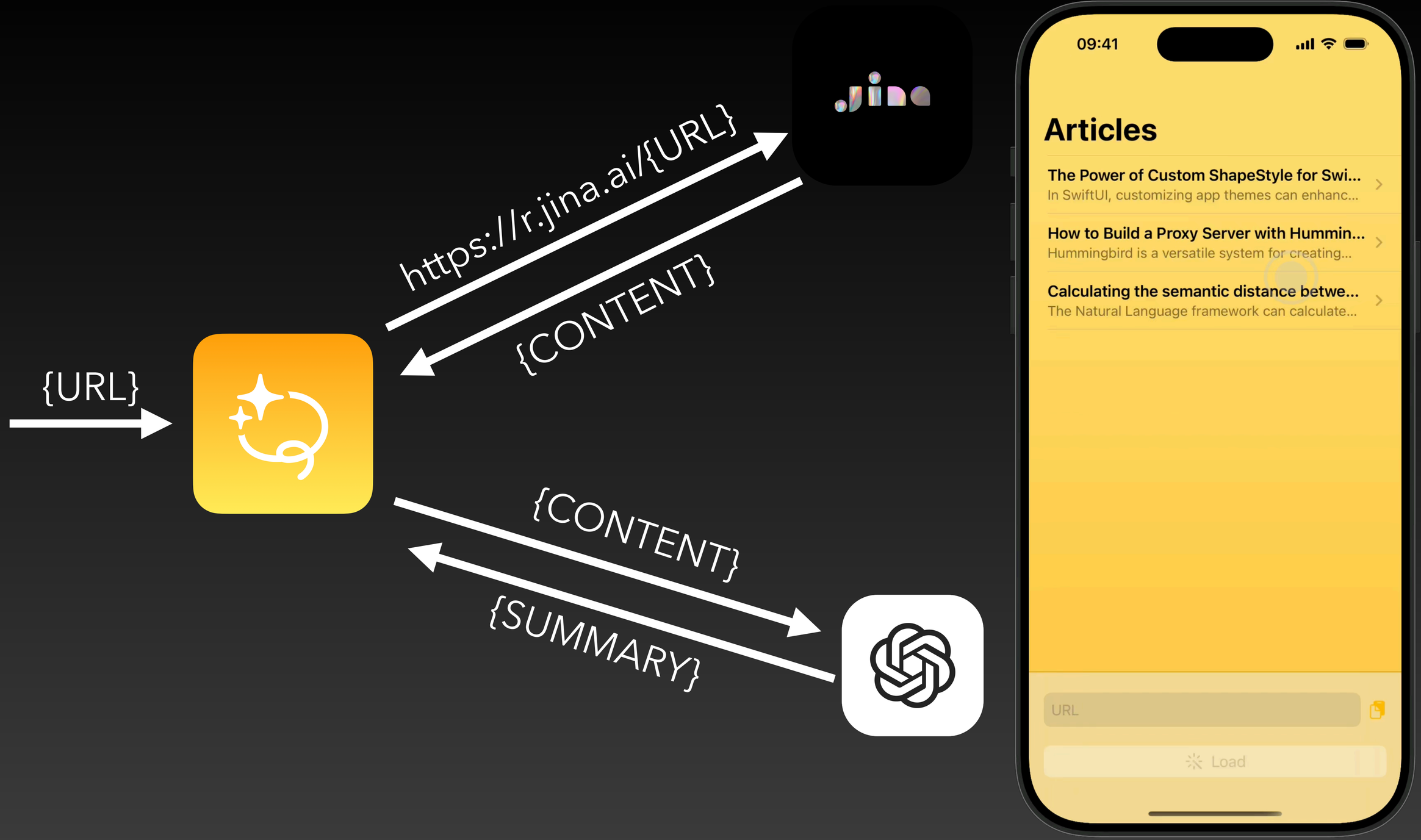
👷 **Not** a backend developer


Your background and experience

Atlas

Atlas


09:41
Articles
The Power of Custom ShapeStyle for Swi...
In SwiftUI, customizing app themes can enhanc...
How to Build a Proxy Server with Hummin...
Hummingbird is a versatile system for creating...
Calculating the semantic distance betwe...
The Natural Language framework can calculate...
URL
Load


https://r.jina.ai/{URL}
{CONTENT}
{URL}
{CONTENT}
{SUMMARY}
09:41
Articles
The Power of Custom ShapeStyle for Swi...
In SwiftUI, customizing app themes can enhanc...
How to Build a Proxy Server with Hummin...
Hummingbird is a versatile system for creating...
Calculating the semantic distance betwe...
The Natural Language framework can calculate...
URL
Load

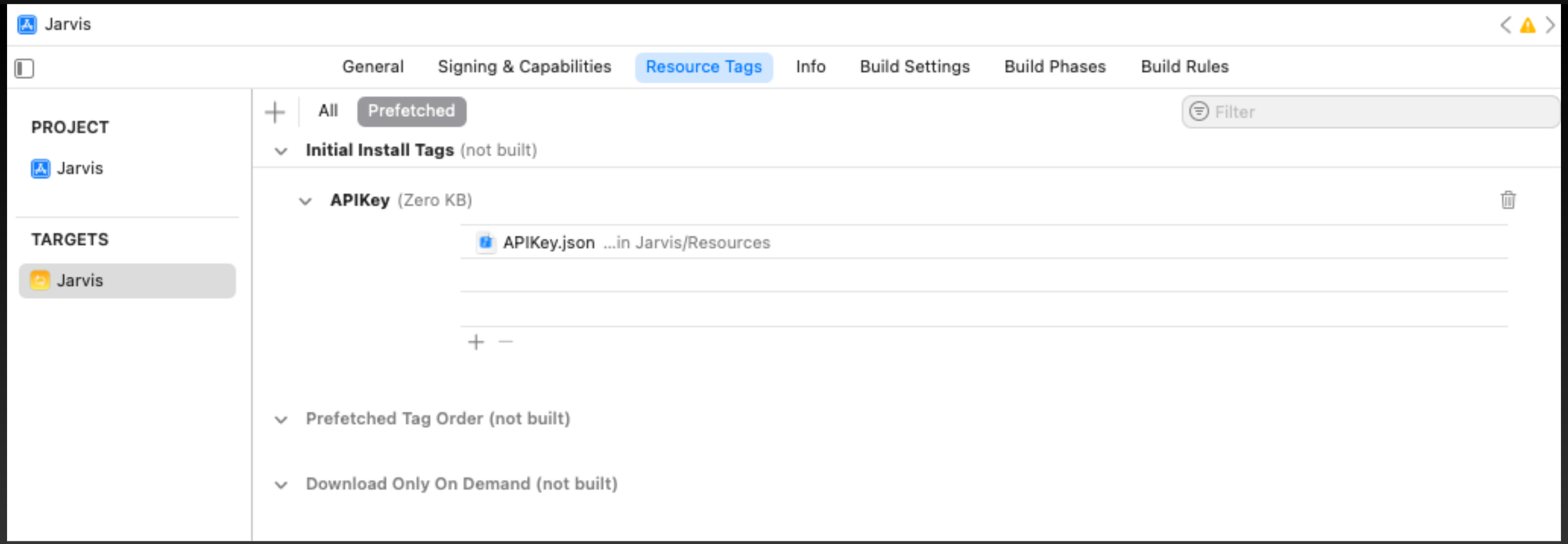
```
var request = URLRequest(url: URL(string: "https://r.jina.ai/")!)
request.httpMethod = "PUT"
request
    .setValue(
        "application/json",
        forHTTPHeaderField: "Content-Type"
    )
request
    .setValue(
        "Bearer jina_a1b2c3d4e5f6g7h8i9j",
        forHTTPHeaderField: "Authorization"
    )
request.httpBody = httpBody
```

```
var request = URLRequest(url: URL(string: "https://r.jina.ai/")!)
request.httpMethod = "PUT"
request
    .setValue(
        "application/json",
        forHTTPHeaderField: "Content-Type"
    )
request
    .setValue(
        "Bearer jina_a1b2c3d4e5f6g7h8i9j",
        forHTTPHeaderField: "Authorization"
    )
request.httpBody = httpBody
```

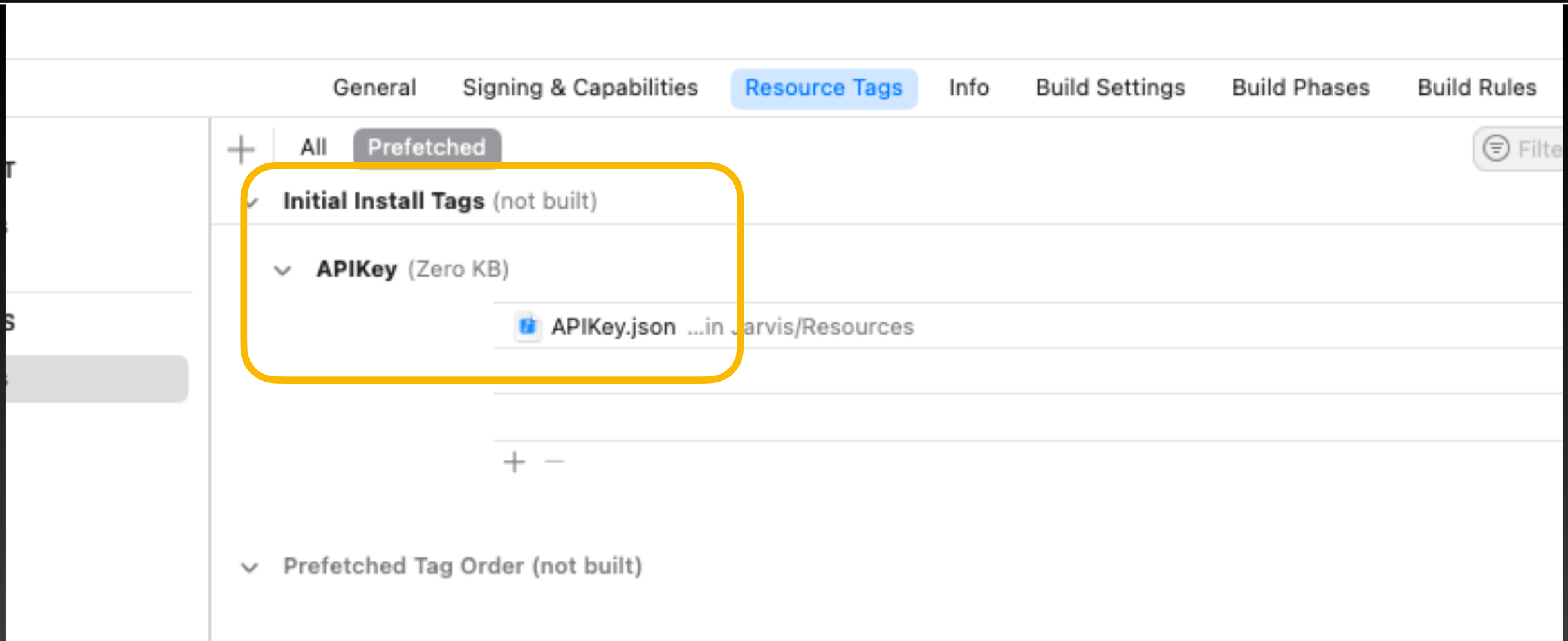
```
enum Constants {
    static let apiKey = "jina_a1b2c3d4e5f6g7h8i9j"
}

var request = URLRequest(url: URL(string: "https://r.jina.ai/")!)
request.httpMethod = "PUT"
request
    .setValue(
        "application/json",
        forHTTPHeaderField: "Content-Type"
    )
request
    .setValue(
        "Bearer \(apiKey)",
        forHTTPHeaderField: "Authorization"
    )
request.httpBody = httpBody
```

```
let apiKey = Bundle.main.object(forInfoDictionaryKey: "API_KEY") as? String ?? „

var request = URLRequest(url: URL(string: "https://r.jina.ai/")!)
request.httpMethod = "PUT"
request
    .setValue(
        "application/json",
        forHTTPHeaderField: "Content-Type"
    )
request
    .setValue(
        "Bearer \(apiKey)",
        forHTTPHeaderField: "Authorization"
    )
request.httpBody = httpBody
```

# On-Demand Resources



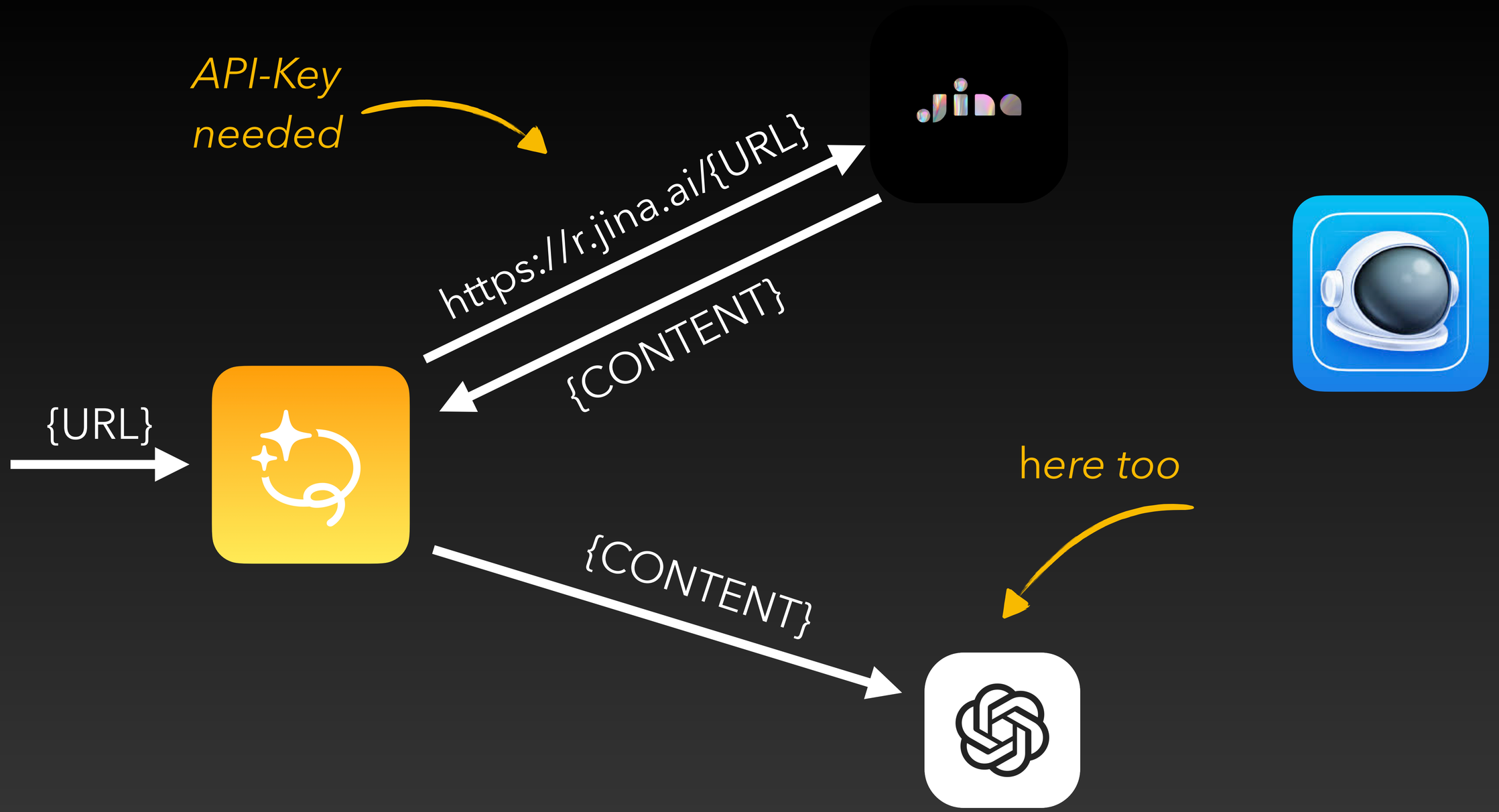
# On-Demand Resources



```
let apiKey = Bundle.main.object(forInfoDictionaryKey: "API_KEY") as? String ?? „

var request = URLRequest(url: URL(string: "https://r.jina.ai/")!)
request.httpMethod = "PUT"
request
    .setValue(
        "application/json",
        forHTTPHeaderField: "Content-Type"
    )
request
    .setValue(
        "Bearer \(apiKey)",
        forHTTPHeaderField: "Authorization"
    )
request.httpBody = httpBody
```

```
func loadApiKey() async throws -> String {
    let request = NSBundleResourceRequest(tags: ["APIKey"])
    try await request.beginAccessingResources()

    let url = Bundle.main.url(forResource: "APIKey", withExtension: "json")!
    let data = try Data(contentsOf: url)

    let decoder = JSONDecoder()
    let apiKeyData = try decoder.decode(ApiKeyData.self, from: data)

    request.endAccessingResources()
    return apiKeyData.apiKey
}

let apiKey = try await loadApiKey()

var request = URLRequest(url: URL(string: "https://r.jina.ai/")!)
request.httpMethod = "PUT"
request
    .setValue(
        "application/json",
        forHTTPHeaderField: "Content-Type"
    )
request
```

```swift
func loadApiKey() async throws -> String {
    let request = NSBundleResourceRequest(tags: ["APIKey"])
    try await request.beginAccessingResources()

    let url = Bundle.main.url(forResource: "APIKey", withExtension: "json")!
    let data = try Data(contentsOf: url)

    let decoder = JSONDecoder()
    let apiKeyData = try decoder.decode(ApiKeyData.self, from: data)

    // TODO: Store in Keychain

    request.endAccessingResources()
    return apiKeyData.apiKey
}

let apiKey = try await loadApiKey()

var request = URLRequest(url: URL(string: "https://r.jina.ai/")!)
request.httpMethod = "PUT"
request
    .setValue(
        "application/json",
        forHTTPHeaderField: "Content-Type"
```


API-Key
needed
https://r.jina.ai/{URL}
{CONTENT}
{URL}
{CONTENT}
here too

# Certificate Pinning

1. Client has public key of Server
2. Handshake between Client & Server (Server sends Certificate to Client)
3. Client check public keys
4. Allows or disallows connection to server

# Backend Development

- No API keys on client-side
- Own request & response models
- Fast backend deployment (compared to app store process)
- Certificate owner
- App Store authorization
- Additional layers (rate limit, input validation etc.)


Swift on Server

Vapor




Swift Server Workgroup (SSWG)

# Swift on Server


Your next project

ServerSide.swift

Hummingbird

Vapor

Hummingbird

Vapor

Hummingbird

Swift

aws

Lambda

Swift Cloud
x
SwiftWasm

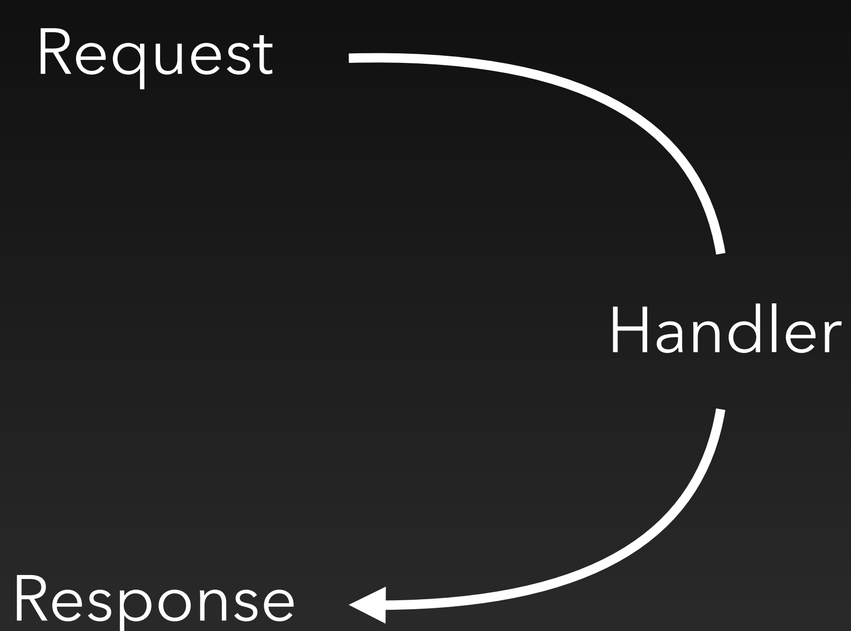

Hummingbird

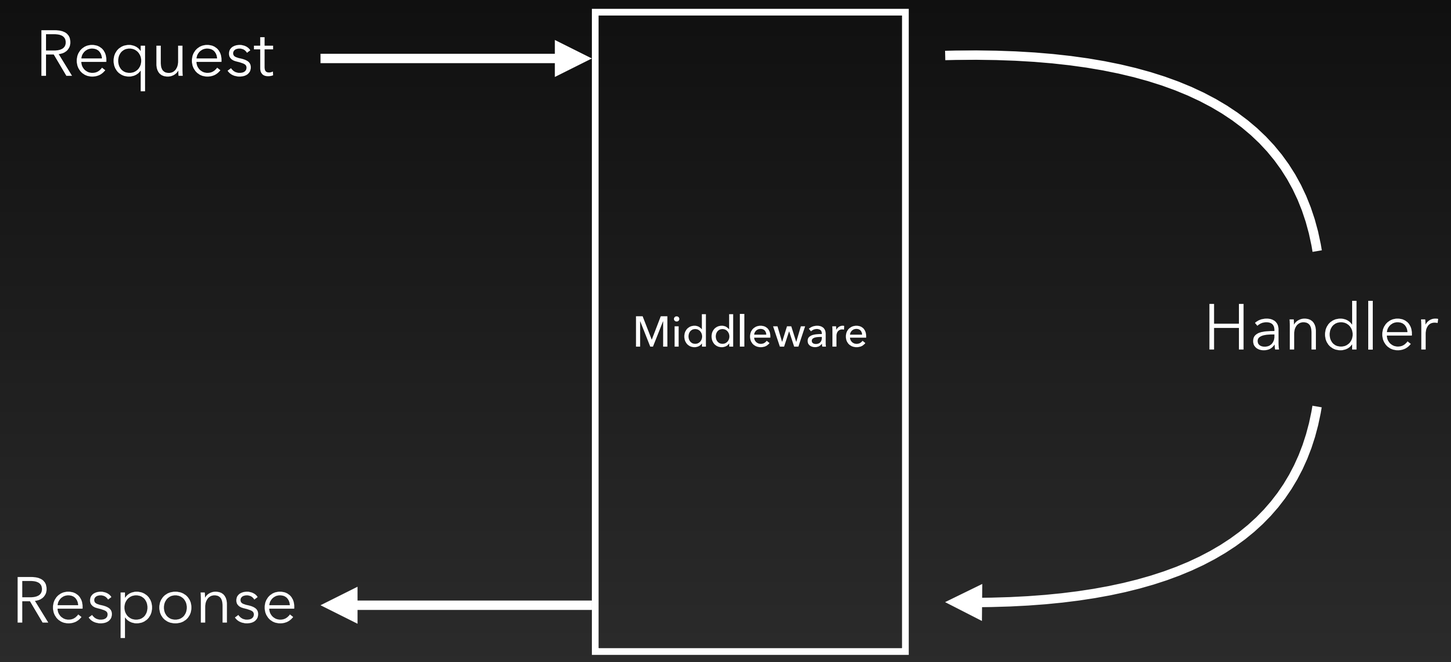
# Hummingbird 2

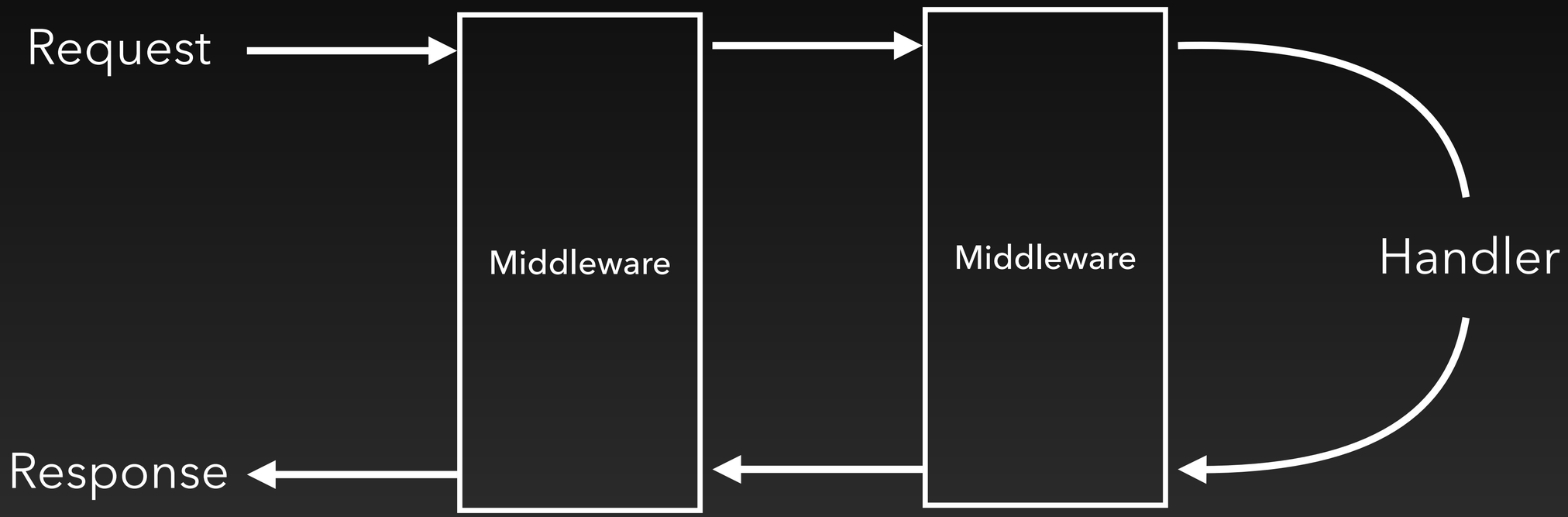
Swift 6

Modular Architecture

Redesigned APIs

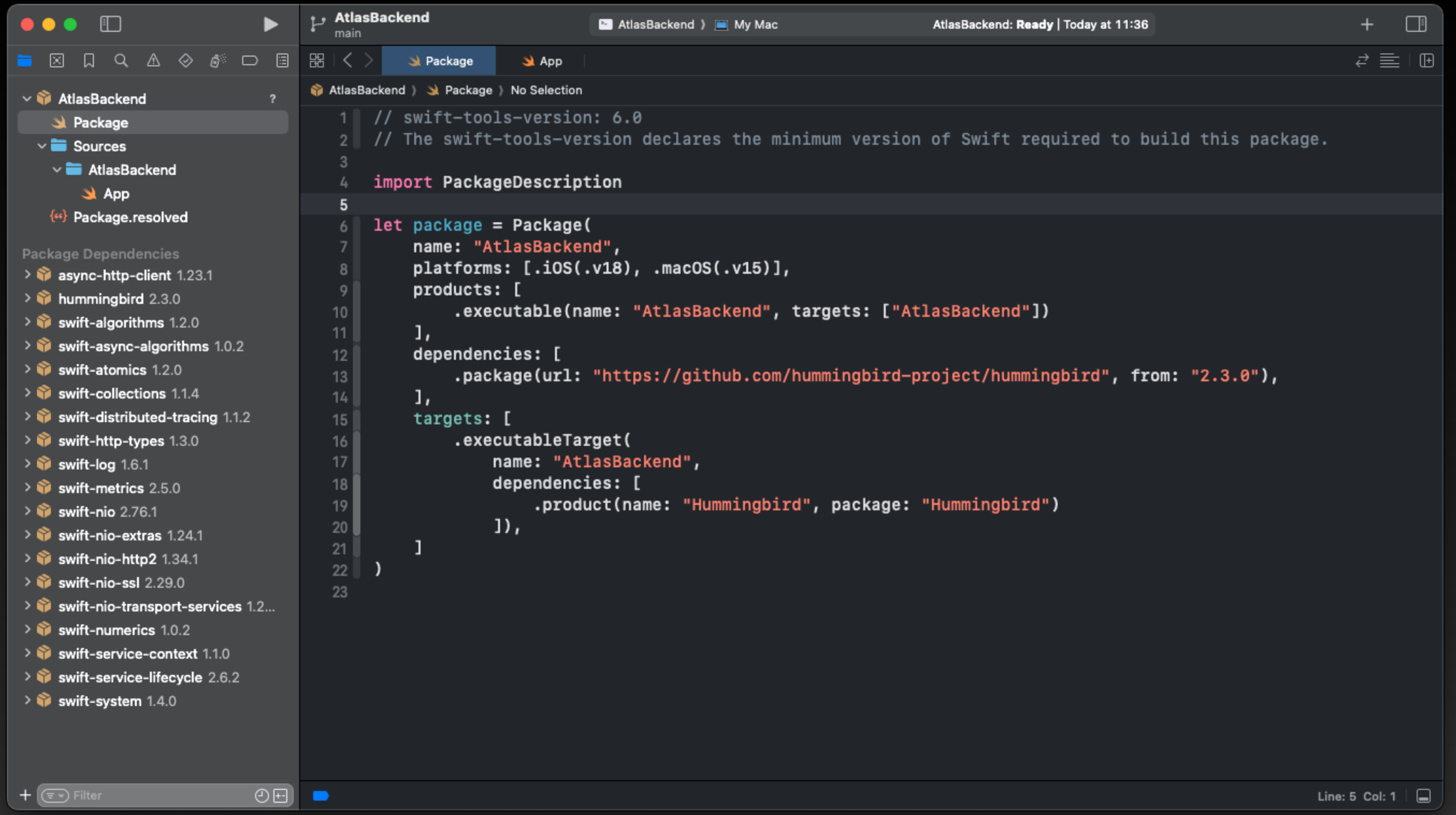

Request
Handler
Response

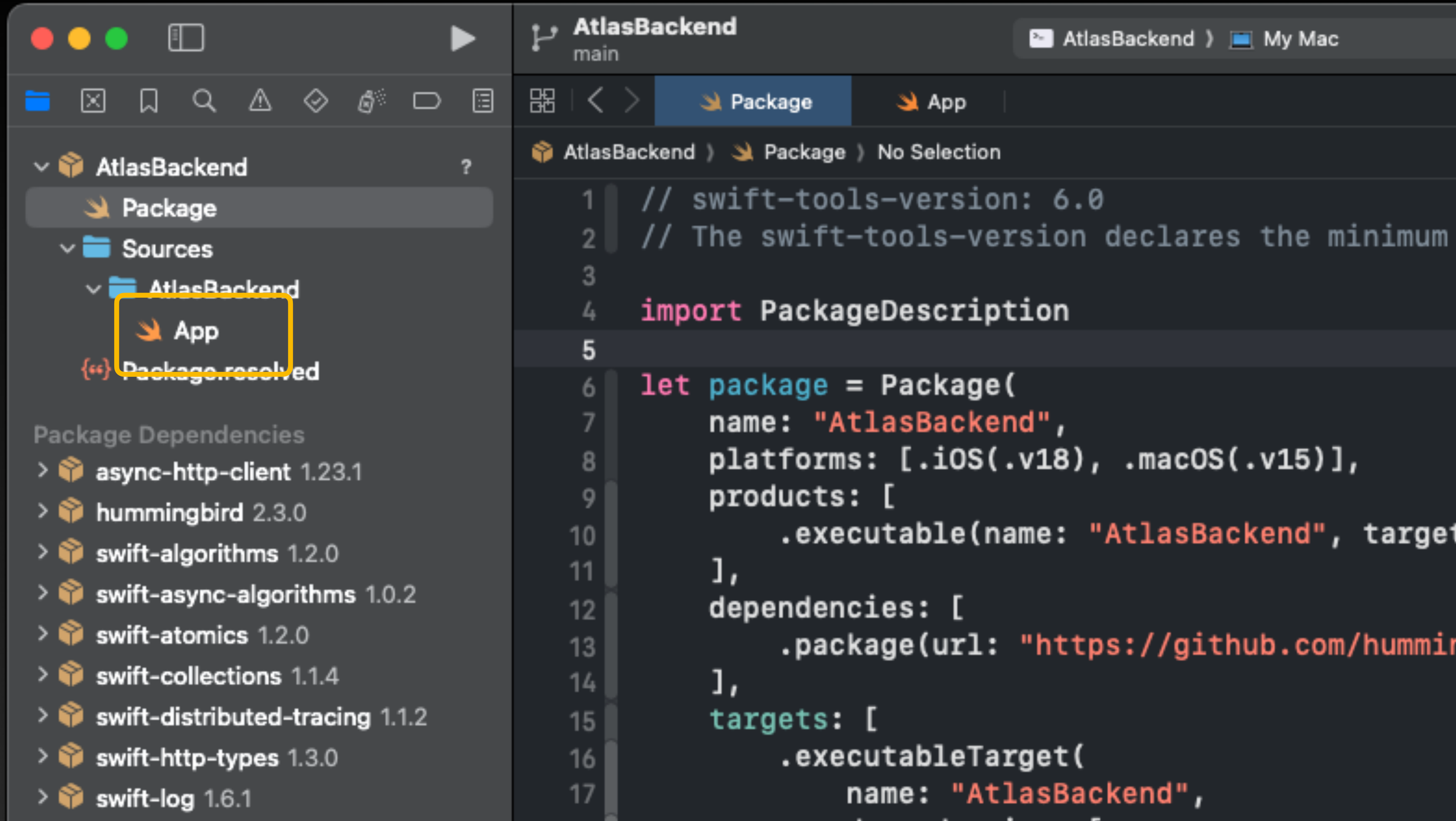

Request
Middleware
Handler
Response


Request
Middleware
Middleware
Handler
Response

# Creating a Hummingbird Project

```
~/workspace/freiwald/Atlas main ❯ mkdir AtlasBackend
~/workspace/freiwald/Atlas main ❯ cd AtlasBackend
~/workspace/freiwald/Atlas/AtlasBackend main ❯ swift package init --type executable
Creating executable package: AtlasBackend
Creating Package.swift
Creating .gitignore
Creating Sources/
Creating Sources/main.swift
~/workspace/freiwald/Atlas/AtlasBackend main ❯ swift package add-dependency https://
github.com/hummingbird-project/hummingbird --from 2.3.0
Updating package manifest at Package.swift... done.
/workspace/freiwald/Atlas/AtlasBackend main ❯ swift package add-target-dependency
Hummingbird AtlasBackend
Updating package manifest at Package.swift... done.
~/workspace/freiwald/Atlas/AtlasBackend main ❯ open Package.swift
```


AtlasBackend
main
AtlasBackend ⟩ My Mac
AtlasBackend: Ready | Today at 11:36
Package
App
AtlasBackend ⟩ Package ⟩ No Selection
AtlasBackend
Package
Sources
AtlasBackend
App
Package.resolved
Package Dependencies
async-http-client 1.23.1
hummingbird 2.3.0
swift-algorithms 1.2.0
swift-async-algorithms 1.0.2
swift-atomics 1.2.0
swift-collections 1.1.4
swift-distributed-tracing 1.1.2
swift-http-types 1.3.0
swift-log 1.6.1
swift-metrics 2.5.0
swift-nio 2.76.1
swift-nio-extras 1.24.1
swift-nio-http2 1.34.1
swift-nio-ssl 2.29.0
swift-nio-transport-services 1.2...
swift-numerics 1.0.2
swift-service-context 1.1.0
swift-service-lifecycle 2.6.2
swift-system 1.4.0
Filter
// swift-tools-version: 6.0
// The swift-tools-version declares the minimum version of Swift required to build this package.
import PackageDescription
let package = Package(
name: "AtlasBackend",
platforms: [.iOS(.v18), .macOS(.v15)],
products: [
.executable(name: "AtlasBackend", targets: ["AtlasBackend"])
],
dependencies: [
.package(url: "https://github.com/hummingbird-project/hummingbird", from: "2.3.0"),
],
targets: [
.executableTarget(
name: "AtlasBackend",
dependencies: [
.product(name: "Hummingbird", package: "Hummingbird")
]),
]
)
Line: 5 Col: 1


AtlasBackend
main
AtlasBackend
My Mac
Package
App
AtlasBackend
Package
No Selection
AtlasBackend
Package
Sources
App
Package Dependencies
async-http-client 1.23.1
hummingbird 2.3.0
swift-algorithms 1.2.0
swift-async-algorithms 1.0.2
swift-atomics 1.2.0
swift-collections 1.1.4
swift-distributed-tracing 1.1.2
swift-http-types 1.3.0
swift-log 1.6.1
// swift-tools-version: 6.0
// The swift-tools-version declares the minimum
import PackageDescription
let package = Package(
name: "AtlasBackend",
platforms: [.iOS(.v18), .macOS(.v15)],
products: [
.executable(name: "AtlasBackend", target
],
dependencies: [
.package(url: "https://github.com/hummin
],
targets: [
.executableTarget(
name: "AtlasBackend",

```swift
@main
struct App: AsyncParsableCommand {
    @Option(name: .shortAndLong)
    var hostname: String = "127.0.0.1"

    @Option(name: .shortAndLong)
    var port: Int = 8080

    func run() async throws {
        let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
        let logger = Logger(label: "AtlasBackend")

        let router = Router(context: BasicRequestContext.self)

        router.get("/") { request, context in
            return "👋 Hello Swift Meetup Munich! 🚀"
        }

        let app = Application(
            router: router,
            configuration: .init(
                address: .hostname(hostname, port: port),
                serverName: "AtlasBackend"
            ),
            eventLoopGroupProvider: .shared(eventLoopGroup),
            logger: logger
        )

        try await app.runService()
    }
}
```

```swift
@main
struct App: AsyncParsableCommand {
    @Option(name: .shortAndLong)
    var hostname: String = "127.0.0.1"

    @Option(name: .shortAndLong)
    var port: Int = 8080

    func run() async throws {
        let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
        let logger = Logger(label: "AtlasBackend")

        let router = Router(context: BasicRequestContext.self)

        router.get("/") { request, context in
            return "👋 Hello Swift Meetup Munich! 🚀"
        }

        let app = Application(
            router: router,
            configuration: .init(
                address: .hostname(hostname, port: port),
                serverName: "AtlasBackend"
            ),
            eventLoopGroupProvider: .shared(eventLoopGroup),
            logger: logger
        )

        try await app.runService()
    }
}
```

```swift
@main
struct App: AsyncParsableCommand {
    @Option(name: .shortAndLong)
    var hostname: String = "127.0.0.1"

    @Option(name: .shortAndLong)
    var port: Int = 8080

    func run() async throws {
        let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
        let logger = Logger(label: "AtlasBackend")

        let router = Router(context: BasicRequestContext.self)

        router.get("/") { request, context in
            return "👋 Hello Swift Meetup Munich! 🚀"
        }

        let app = Application(
            router: router,
            configuration: .init(
                address: .hostname(hostname, port: port),
                serverName: "AtlasBackend"
            ),
            eventLoopGroupProvider: .shared(eventLoopGroup),
            logger: logger
        )

        try await app.runService()
    }
}
```

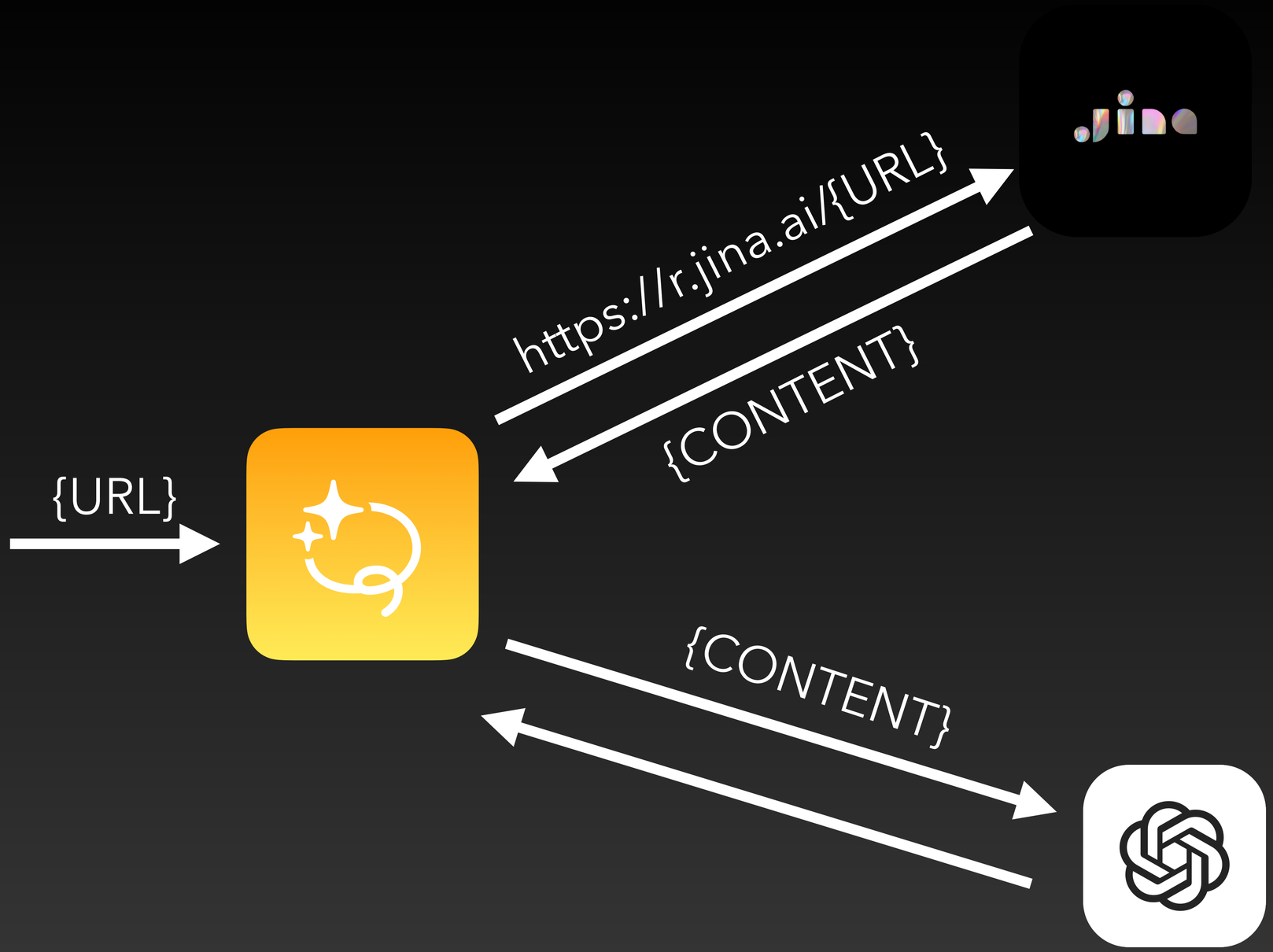
```swift
@main
struct App: AsyncParsableCommand {
    @Option(name: .shortAndLong)
    var hostname: String = "127.0.0.1"

    @Option(name: .shortAndLong)
    var port: Int = 8080

    func run() async throws {
        let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
        let logger = Logger(label: "AtlasBackend")

        let router = Router(context: BasicRequestContext.self)

        router.get("/") { request, context in
            return "👋 Hello Swift Meetup Munich! 🚀"
        }

        let app = Application(
            router: router,
            configuration: .init(
                address: .hostname(hostname, port: port),
                serverName: "AtlasBackend"
            ),
            eventLoopGroupProvider: .shared(eventLoopGroup),
            logger: logger
        )

        try await app.runService()
    }
}
```

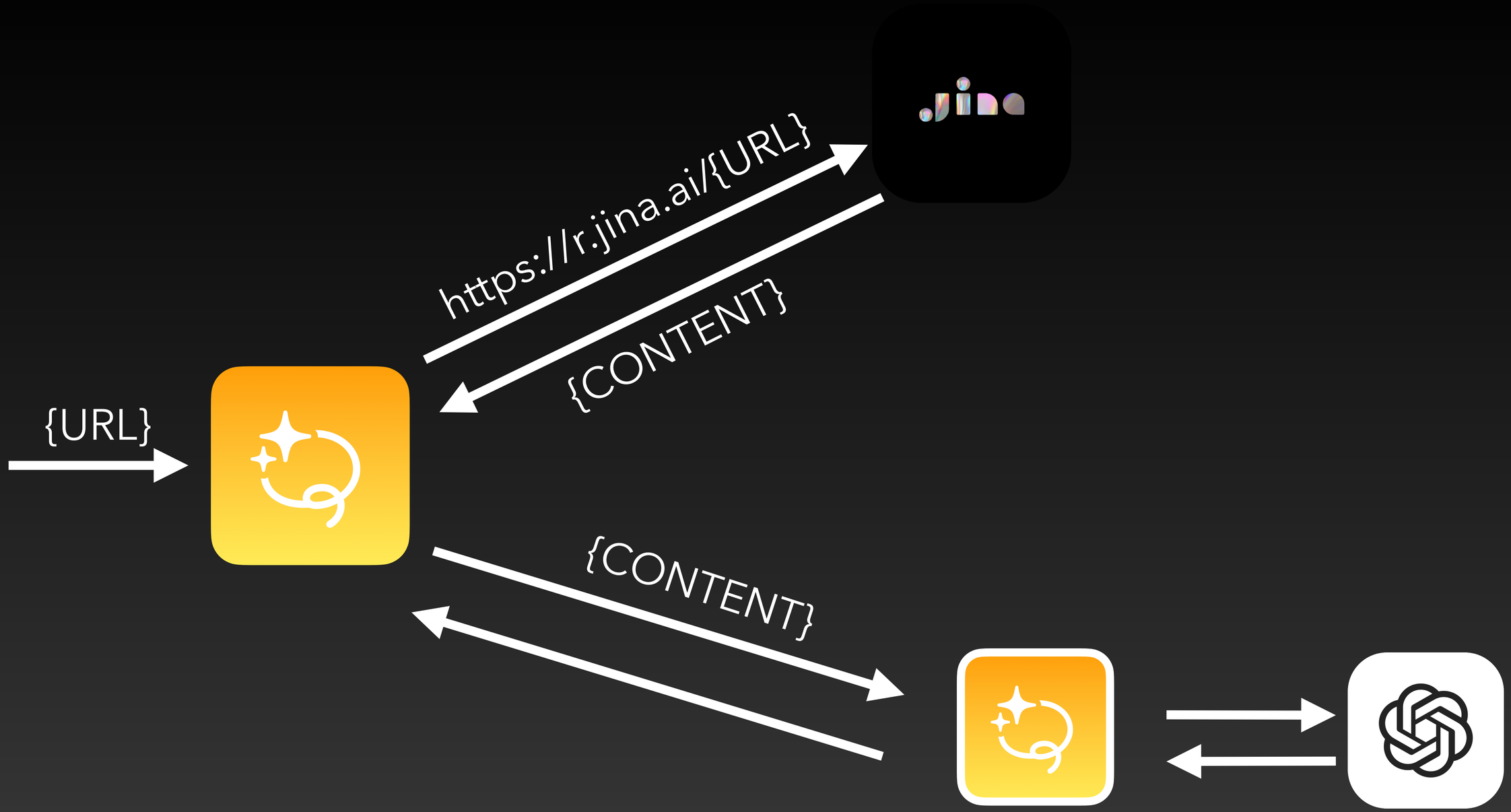
```swift
@main
struct App: AsyncParsableCommand {
    @Option(name: .shortAndLong)
    var hostname: String = "127.0.0.1"

    @Option(name: .shortAndLong)
    var port: Int = 8080

    func run() async throws {
        let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
        let logger = Logger(label: "AtlasBackend")

        let router = Router(context: BasicRequestContext.self)

        router.get("/") { request, context in
            return "👋 Hello Swift Meetup Munich! 🚀"
        }

        let app = Application(
            router: router,
            configuration: .init(
                address: .hostname(hostname, port: port),
                serverName: "AtlasBackend"
            ),
            eventLoopGroupProvider: .shared(eventLoopGroup),
            logger: logger
        )

        try await app.runService()
    }
}
```

```swift
@main
struct App: AsyncParsableCommand {
    @Option(name: .shortAndLong)
    var hostname: String = "127.0.0.1"

    @Option(name: .shortAndLong)
    var port: Int = 8080

    func run() async throws {
        let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
        let logger = Logger(label: "AtlasBackend")

        let router = Router(context: BasicRequestContext.self)

        router.get("/") { request, context in
            return "👋 Hello Swift Meetup Munich! 🚀"
        }

        let app = Application(
            router: router,
            configuration: .init(
                address: .hostname(hostname, port: port),
                serverName: "AtlasBackend"
            ),
            eventLoopGroupProvider: .shared(eventLoopGroup),
            logger: logger
        )

        try await app.runService()
    }
}
```

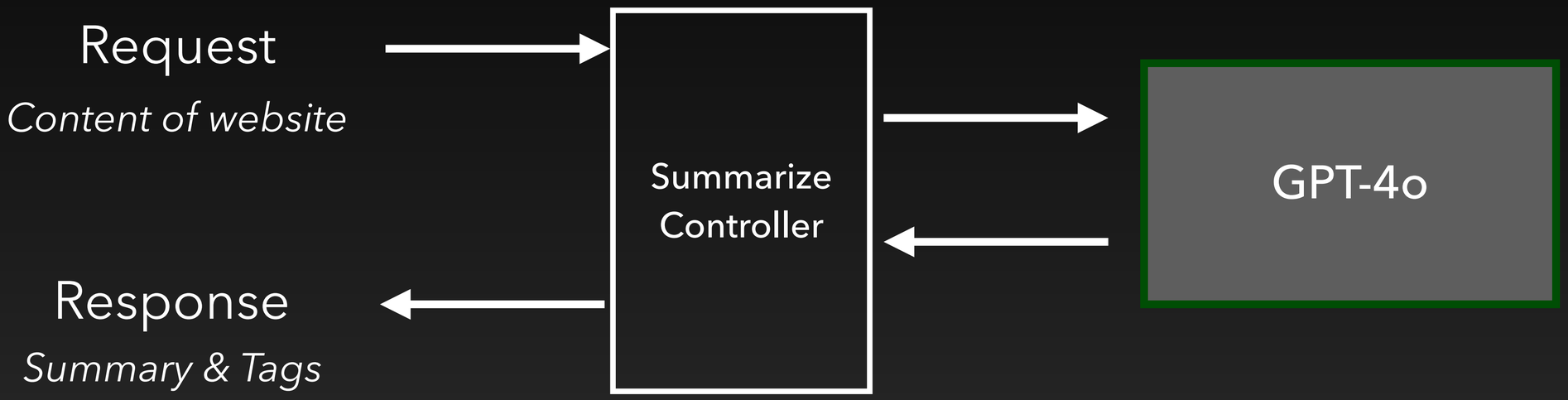
```swift
@main
struct App: AsyncParsableCommand {
    @Option(name: .shortAndLong)
    var hostname: String = "127.0.0.1"

    @Option(name: .shortAndLong)
    var port: Int = 8080

    func run() async throws {
        let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
        let logger = Logger(label: "AtlasBackend")

        let router = Router(context: BasicRequestContext.self)

        router.get("/") { request, context in
            return "👋 Hello Swift Meetup Munich! 🚀"
        }

        let app = Application(
            router: router,
            configuration: .init(
                address: .hostname(hostname, port: port),
                serverName: "AtlasBackend"
            ),
            eventLoopGroupProvider: .shared(eventLoopGroup),
            logger: logger
        )

        try await app.runService()
    }
}
```

# Coding


https://r.jina.ai/{URL}
{CONTENT}
{URL}
{CONTENT}


https://r.jina.ai/{URL}
{CONTENT}
{URL}
{CONTENT}

```swift
public struct SummaryRequest: Sendable, Codable {
    public let content: String

    public init(content: String) {
        self.content = content
    }
}
```




```swift
public struct SummaryResponse: Sendable, Codable {
    public let summary: String
    public let tags: [String]

    package init(summary: String, tags: [String]) {
        self.summary = summary
        self.tags = tags
    }
}
```

```
struct SummaryController<Context: RequestContext> {
}
```

```
struct SummaryController<Context: RequestContext> {
    func addRoutes(to group: RouterGroup<Context>) {
        group.post(use: summarize)
    }
}
```

```
struct SummaryController<Context: RequestContext> {
    func addRoutes(to group: RouterGroup<Context>) {
        group.post(use: summarize)
    }
}
```

```swift
struct SummaryController<Context: RequestContext> {
    func addRoutes(to group: RouterGroup<Context>) {
        group.post(use: summarize)
    }

    @Sendable private func summarize(
        _ request: Request,
        context: Context
    ) async throws -> SummaryResponse {
        let input = try await request.decode(
            as: SummaryRequest.self,
            context: context
        )

        context.logger.info("Receive content: \(input.content.prefix(50))")

        let response = SummaryResponse(summary: "TODO: Summary", tags: [])

        return response
    }
}
```

```swift
struct SummaryController<Context: RequestContext> {
    func addRoutes(to group: RouterGroup<Context>) {
        group.post(use: summarize)
    }

    @Sendable private func summarize(
        _ request: Request,
        context: Context
    ) async throws -> SummaryResponse {
        let input = try await request.decode(
            as: SummaryRequest.self,
            context: context
        )

        context.logger.info("Receive content: \(input.content.prefix(50))")

        let response = SummaryResponse(summary: "TODO: Summary", tags: [])

        return response
    }
}
```

```swift
struct SummaryController<Context: RequestContext> {
    func addRoutes(to group: RouterGroup<Context>) {
        group.post(use: summarize)
    }

    @Sendable private func summarize(
        _ request: Request,
        context: Context
    ) async throws -> SummaryResponse {
        let input = try await request.decode(
            as: SummaryRequest.self,
            context: context
        )

        context.logger.info("Receive content: \(input.content.prefix(50))")

        let response = SummaryResponse(summary: "TODO: Summary", tags: [])

        return response
    }
}
```

```
protocol SummarizserService: Sendable {
    func summarizse(_ request: SummaryRequest) async throws -> SummaryResponse
}
```

```swift
protocol SummarizserService: Sendable {
    func summarizse(_ request: SummaryRequest) async throws -> SummaryResponse
}
struct OpenAISummarizser: SummarizserService {
    private let service: any OpenAIService
    private let schema: JSONSchemaResponseFormat = { ••• }()

    init(resourceName: String, apiKey: String) {
        service = OpenAIServiceFactory.service(
            azureConfiguration: AzureOpenAIConfiguration(
                resourceName: resourceName,
                openAIAPIKey: .apiKey(apiKey),
                apiVersion: "2024-10-01-preview"
            )
        )
    }

    let systemMessage = { ••• }()

    func summarizse(_ request: SummaryRequest) async throws -> SummaryResponse { ••• }
}
```

```
    }

    let systemMessage = { ••• }()

    func summarizse(_ request: SummaryRequest) async throws -> SummaryResponse {









    }
}
```

```
    }

    let systemMessage = { ••• }()

    func summarizse(_ request: SummaryRequest) async throws -> SummaryResponse {
        let result = try await service.startChat(
            parameters: .init(
                messages: [
                    .init(role: .system, content: .text(systemMessage)),
                    .init(role: .user, content: .text(request.content))
                ],
                model: model,
                responseFormat: .jsonSchema(schema)
            )
        )

        guard
            let content = result.choices.first?.message.content,
            let data = content.data(using: .utf8)
        else { throw SummarizserError.noChatOutput }

        let decoded = try JSONDecoder().decode(SummaryResponse.self, from: data)
        return decoded
    }
}
```

```
    }

    let systemMessage = { ••• }()

    func summarizse(_ request: SummaryRequest) async throws -> SummaryResponse {
        let result = try await service.startChat(
            parameters: .init(
                messages: [
                    .init(role: .system, content: .text(systemMessage)),
                    .init(role: .user, content: .text(request.content))
                ],
                model: model,
                responseFormat: .jsonSchema(schema)
            )
        )

        guard
            let content = result.choices.first?.message.content,
            let data = content.data(using: .utf8)
        else { throw SummarizserError.noChatOutput }

        let decoded = try JSONDecoder().decode(SummaryResponse.self, from: data)
        return decoded
    }
}
```

```
    }

    let systemMessage = { ••• }()

    func summarizse(_ request: SummaryRequest) async throws -> SummaryResponse {
        let result = try await service.startChat(
            parameters: .init(
                messages: [
                    .init(role: .system, content: .text(systemMessage)),
                    .init(role: .user, content: .text(request.content))
                ],
                model: model,
                responseFormat: .jsonSchema(schema)
            )
        )

        guard
            let content = result.choices.first?.message.content,
            let data = content.data(using: .utf8)
        else { throw SummarizserError.noChatOutput }

        let decoded = try JSONDecoder().decode(SummaryResponse.self, from: data)
        return decoded
    }
}
```

```
    }

    let systemMessage = { ••• }()

    func summarizse(_ request: SummaryRequest) async throws -> SummaryResponse {
        let result = try await service.startChat(
            parameters: .init(
                messages: [
                    .init(role: .system, content: .text(systemMessage)),
                    .init(role: .user, content: .text(request.content))
                ],
                model: model,
                responseFormat: .jsonSchema(schema)
            )
        )

        guard
            let content = result.choices.first?.message.content,
            let data = content.data(using: .utf8)
        else { throw SummarizserError.noChatOutput }

        let decoded = try JSONDecoder().decode(SummaryResponse.self, from: data)
        return decoded
    }
}
```

```swift
struct SummaryController<Context: RequestContext> {
    func addRoutes(to group: RouterGroup<Context>) {
        group.post(use: summarize)
    }

    @Sendable private func summarize(
        _ request: Request,
        context: Context
    ) async throws -> SummaryResponse {
        let input = try await request.decode(
            as: SummaryRequest.self,
            context: context
        )

        context.logger.info("Receive content: \(input.content.prefix(50))")

        let response = SummaryResponse(summary: "TODO: Summary", tags: [])

        return response
    }
}
```

```swift
struct SummaryController<Context: RequestContext> {
    let service: any SummarizserService

    func addRoutes(to group: RouterGroup<Context>) {
        group.post(use: summarize)
    }

    @Sendable private func summarize(
        _ request: Request,
        context: Context
    ) async throws -> SummaryResponse {
        let input = try await request.decode(
            as: SummaryRequest.self,
            context: context
        )

        context.logger.info("Receive content: \(input.content.prefix(50))")

        let response = SummaryResponse(summary: "TODO: Summary", tags: [])

        return response
    }
}
```

```swift
struct SummaryController<Context: RequestContext> {
    let service: any SummarizserService

    func addRoutes(to group: RouterGroup<Context>) {
        group.post(use: summarize)
    }

    @Sendable private func summarize(
        _ request: Request,
        context: Context
    ) async throws -> SummaryResponse {
        let input = try await request.decode(
            as: SummaryRequest.self,
            context: context
        )

        context.logger.info("Receive content: \(input.content.prefix(50))")

        let response = try await service.summarizse(input)

        return response
    }
}
```

```swift
func run() async throws {
    let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
    let logger = Logger(label: "AtlasBackend")

    let router = Router(context: BasicRequestContext.self)

    router.get("/") { request, context in
        let name = request.uri.queryParameters.get("name") ?? "Swift Meetup Munich"
        return "👋 Hello \(name)! 🚀 \n\nsummarizse Endpoint is not available anymore 😉"
    }

    let app = Application(
        router: router,
        configuration: .init(
            address: .hostname(hostname, port: port),
            serverName: "AtlasBackend"
        ),
        eventLoopGroupProvider: .shared(eventLoopGroup),
        logger: logger
    )

    try await app.runService()
}
```

```swift
func run() async throws {
    let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
    let logger = Logger(label: "AtlasBackend")

    let router = Router(context: BasicRequestContext.self)

    router.get("/") { request, context in
        let name = request.uri.queryParameters.get("name") ?? "Swift Meetup Munich"
        return "👋 Hello \(name)! 🚀 \n\nsummarizse Endpoint is not available anymore 😉"
    }

    SummaryController(service: any SummarizserService)
        .addRoutes(to: router.group("summarizse"))

    let app = Application(
        router: router,
        configuration: .init(
            address: .hostname(hostname, port: port),
            serverName: "AtlasBackend"
        ),
        eventLoopGroupProvider: .shared(eventLoopGroup),
        logger: logger
    )

    try await app.runService()
```

```swift
func run() async throws {
    let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
    let logger = Logger(label: "AtlasBackend")

    let router = Router(context: BasicRequestContext.self)

    router.get("/") { request, context in
        let name = request.uri.queryParameters.get("name") ?? "Swift Meetup Munich"
        return "👋 Hello \(name)! 🚀 \n\nsummarizse Endpoint is not available anymore 😉"
    }

    SummaryController(service: any SummarizserService)
        .addRoutes(to: router.group("summarizse"))

    let app = Application(
        router: router,
        configuration: .init(
            address: .hostname(hostname, port: port),
            serverName: "AtlasBackend"
        ),
        eventLoopGroupProvider: .shared(eventLoopGroup),
        logger: logger
    )

    try await app.runService()
```

*e.g. https://atlas-backend.dev/summarizse*

```swift
func run() async throws {
    let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
    let logger = Logger(label: "AtlasBackend")

    let router = Router(context: BasicRequestContext.self)

    router.get("/") { request, context in
        let name = request.uri.queryParameters.get("name") ?? "Swift Meetup Munich"
        return "👋 Hello \(name)! 🚀 \n\nsummarizse Endpoint is not available anymore 😉"
    }

    let summarizserService: SummarizserService = OpenAISummarizser(
        resourceName: "",
        apiKey: ""
    )

    SummaryController(service: any SummarizserService)
        .addRoutes(to: router.group("summarizse"))

    let app = Application(
        router: router,
        configuration: .init(
            address: .hostname(hostname, port: port),
            serverName: "AtlasBackend"
        )
```

```swift
func run() async throws {
    let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
    let logger = Logger(label: "AtlasBackend")

    let router = Router(context: BasicRequestContext.self)

    router.get("/") { request, context in
        let name = request.uri.queryParameters.get("name") ?? "Swift Meetup Munich"
        return "👋 Hello \(name)! 🚀 \n\nsummarizse Endpoint is not available anymore 😉"
    }

    let summarizserService: SummarizserService = OpenAISummarizser(
        resourceName: "",
        apiKey: ""
    )

    SummaryController(service: summarizserService)
        .addRoutes(to: router.group("summarizse"))

    let app = Application(
        router: router,
        configuration: .init(
            address: .hostname(hostname, port: port),
            serverName: "AtlasBackend"
        )
```

```swift
func run() async throws {
    let eventLoopGroup = MultiThreadedEventLoopGroup.singleton
    let logger = Logger(label: "AtlasBackend")

    let router = Router(context: BasicRequestContext.self)

    router.get("/") { request, context in
        let name = request.uri.queryParameters.get("name") ?? "Swift Meetup Munich"
        return "👋 Hello \(name)! 🚀 \n\nsummarizse Endpoint is not available anymore 😉
    }

    let (apiKey, resourceName) = try await loadEnv(logger)

    let summarizserService: SummarizserService = OpenAISummarizser(
        resourceName:resourceName,
        apiKey: apiKey
    )

    SummaryController(service: summarizserService)
        .addRoutes(to: router.group("summarizse"))

    let app = Application(
        router: router,
        configuration: .init(
```

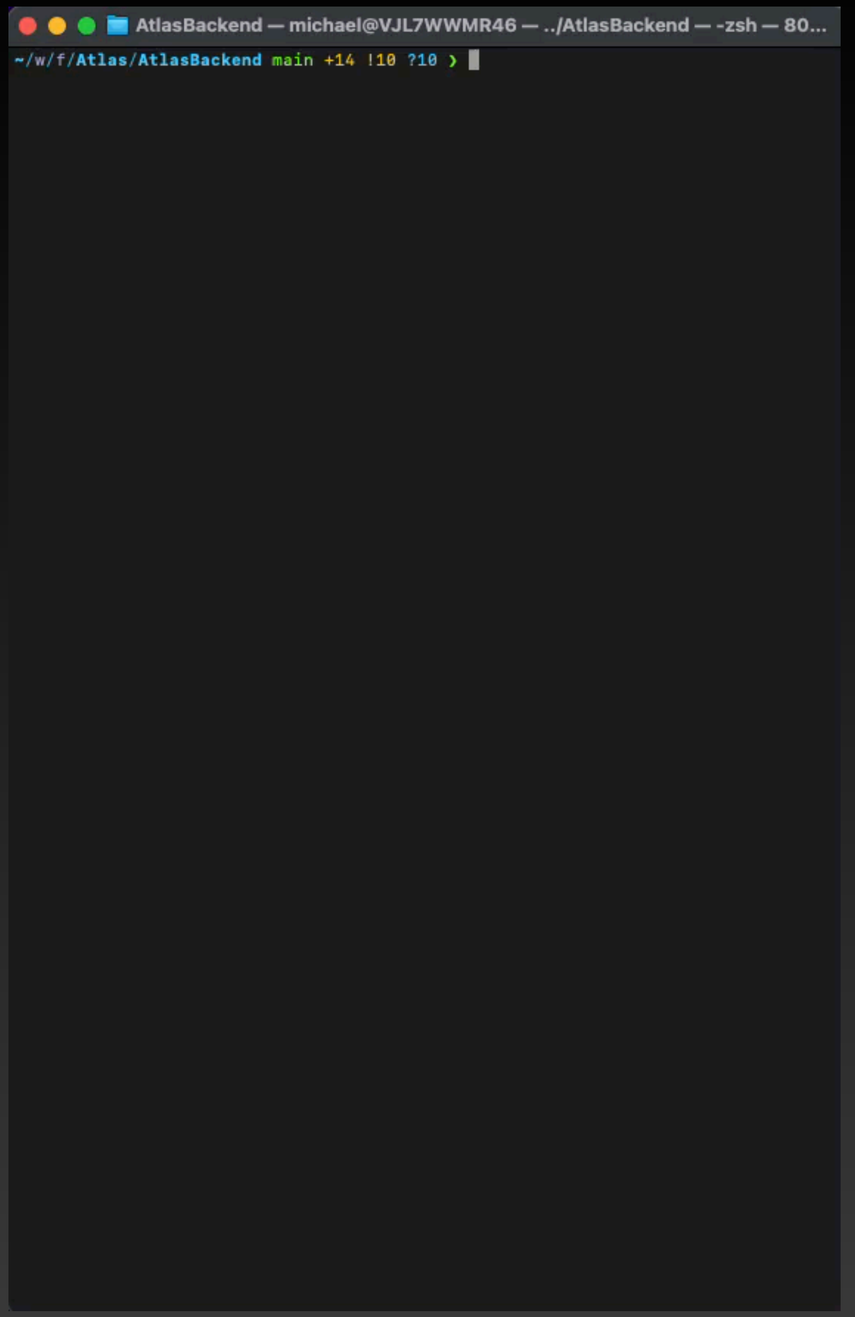
# Demo

# And now?


Fly.io


Fly.io

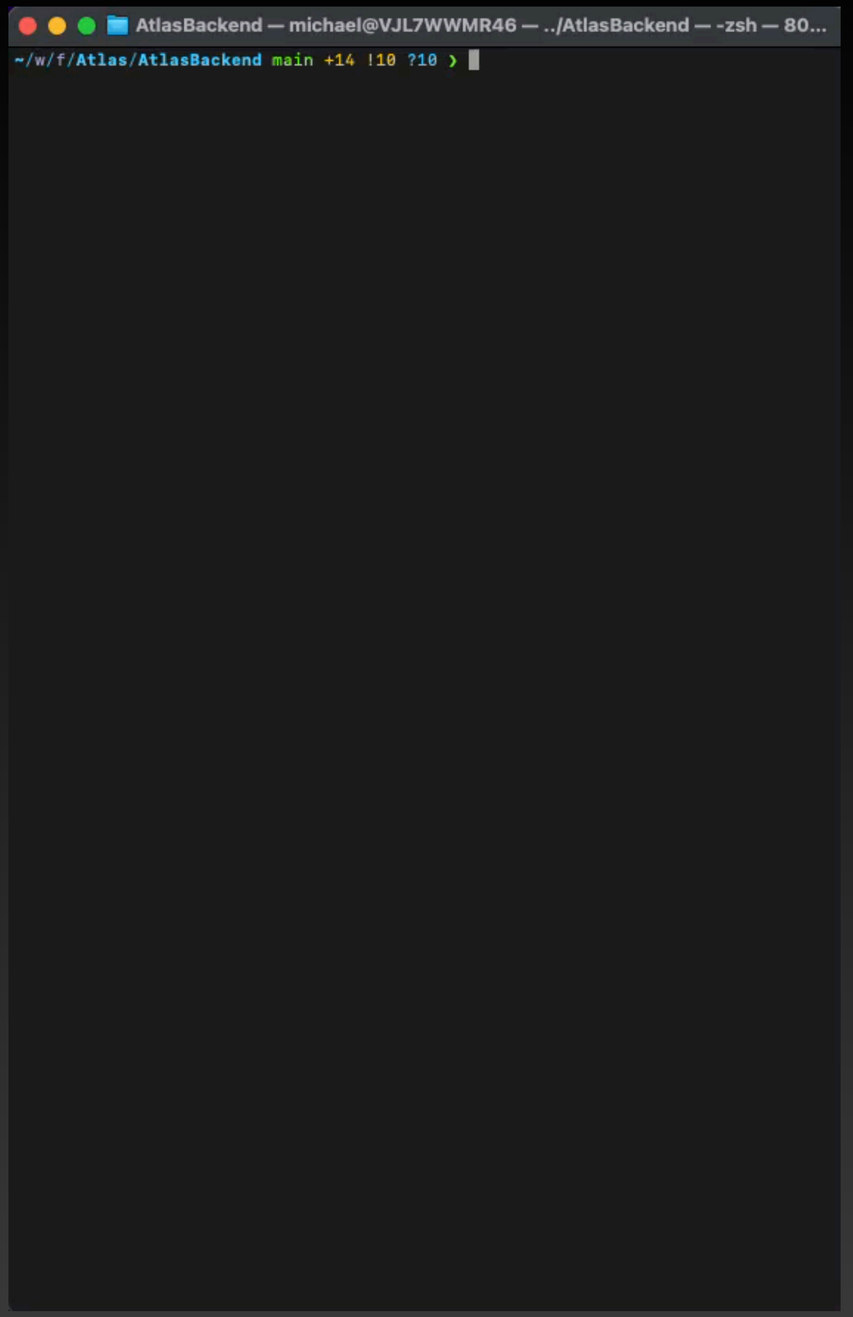

2x


AtlasBackend — michael@VJL7WWMR46 — ../AtlasBackend — -zsh — 80...
~/w/f/Atlas/AtlasBackend main +14 !10 ?10 ❯


Fly.io
Dockerfiles and fly.toml

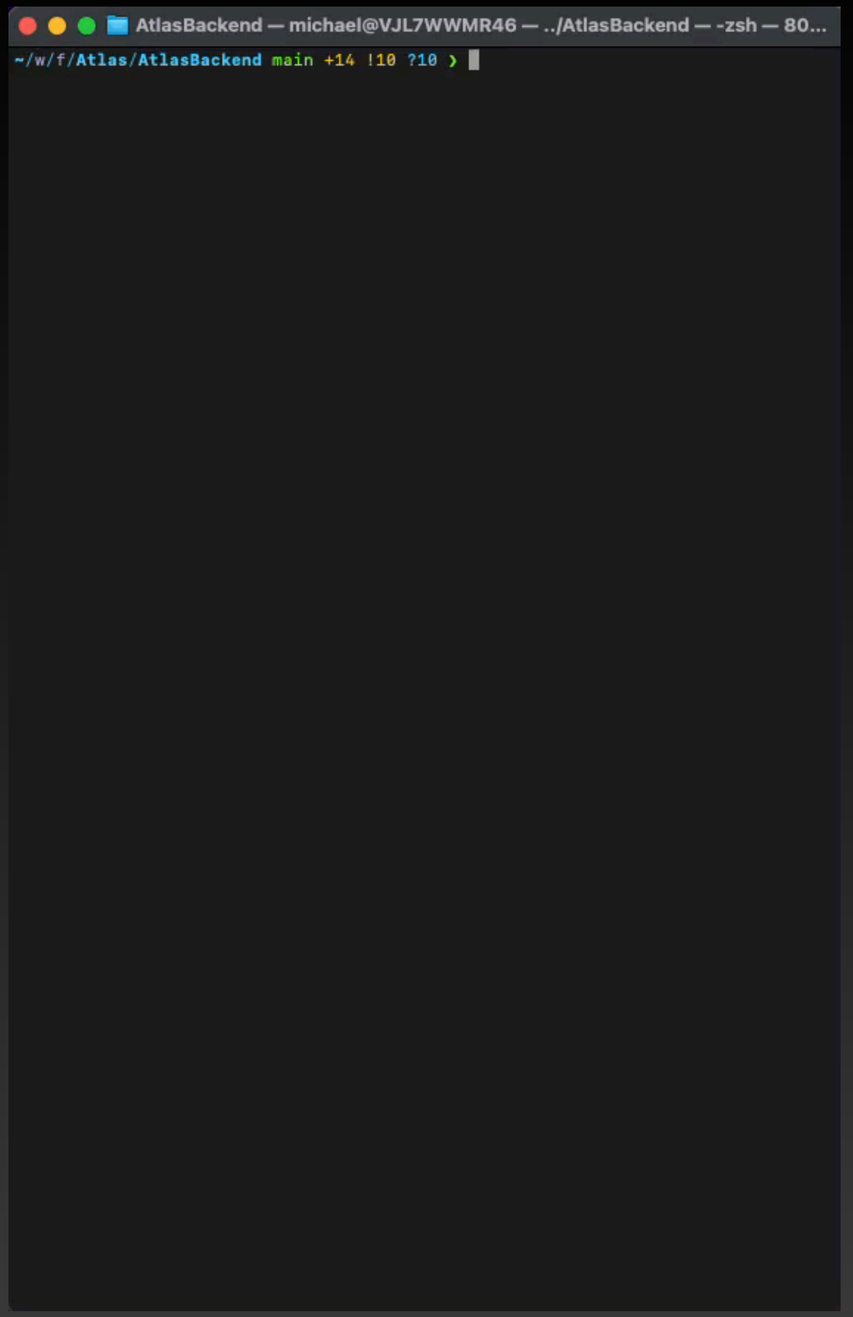
# Dockerfile

- Isolated container with running Linux
- Image with Swift compiler installed
- Builds our backend application
- Access to environment variables
- Runs application

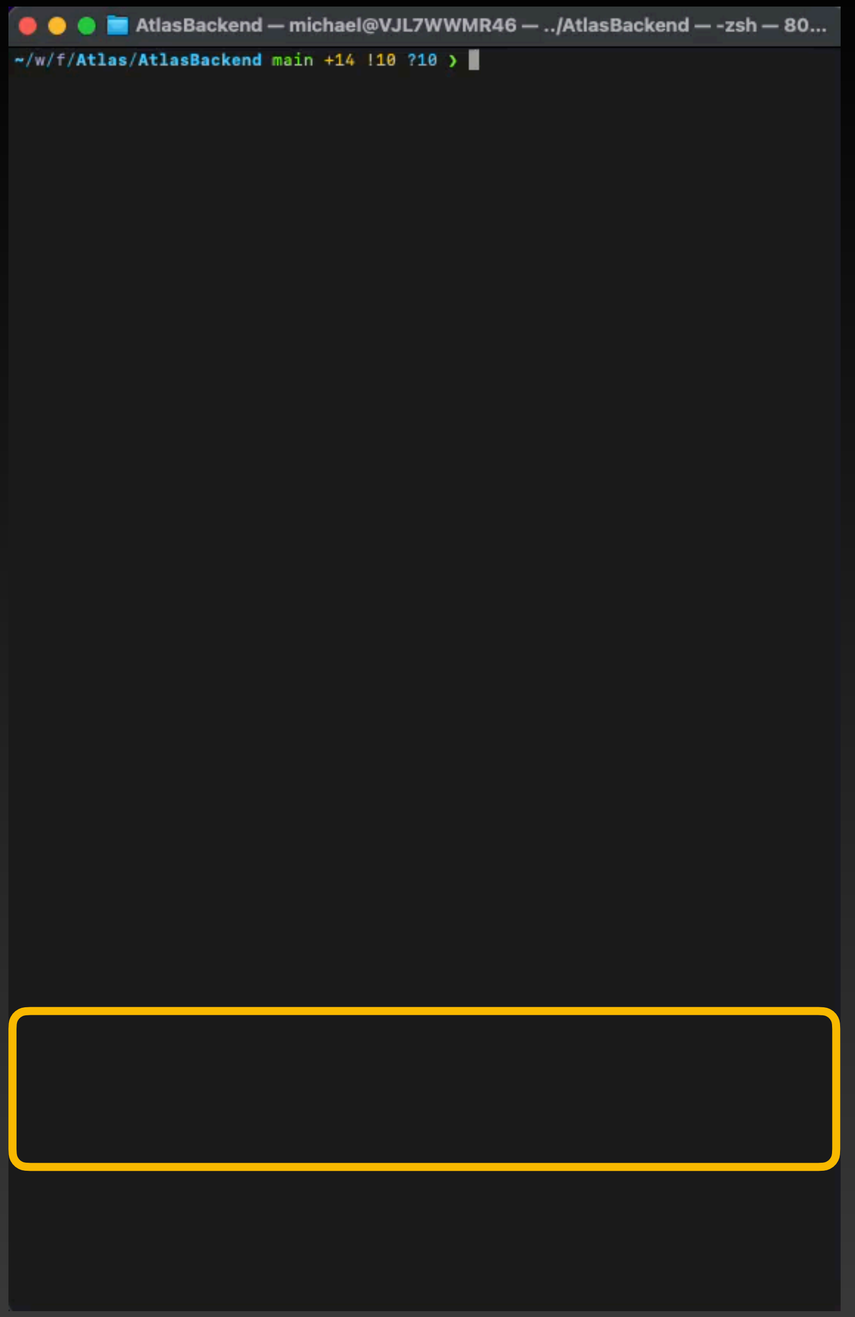


# fly.toml

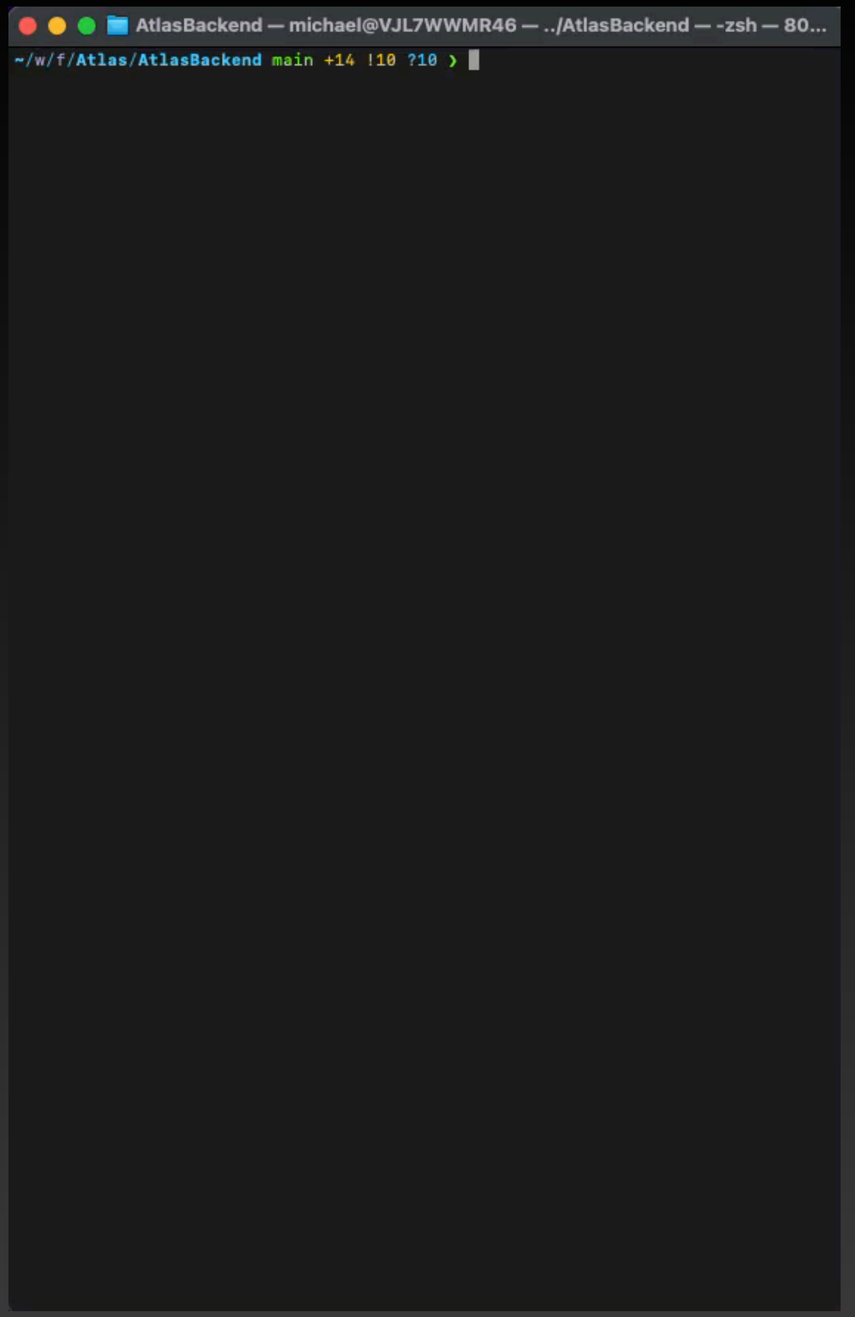
- Configures the fly app
- Defines the number of fly machines
- Defines Resources



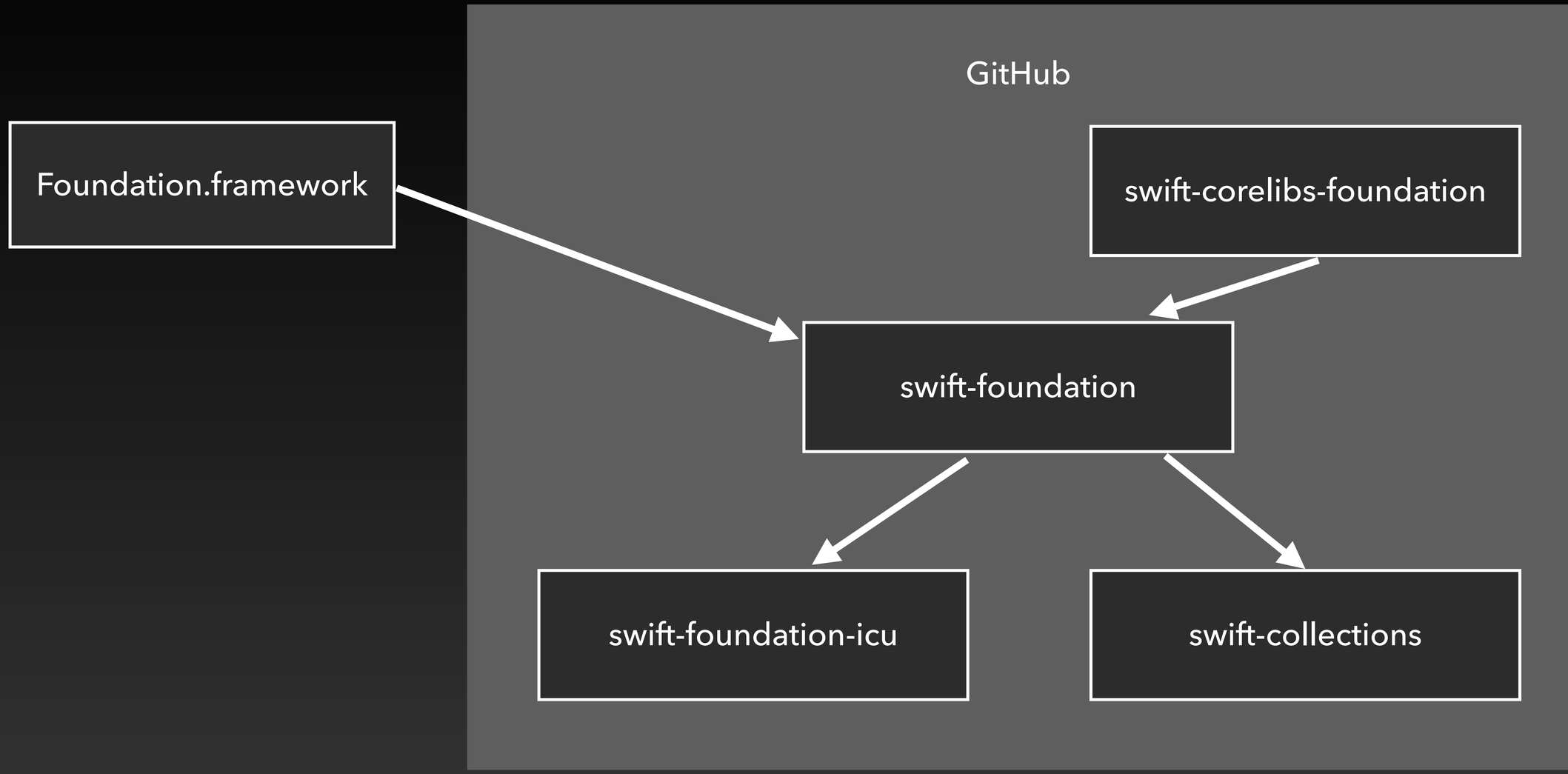
❌ Error: no such module ‘OSLog’



❌ Error: 'URLSessionConfiguration' is unavailable: This type has moved to the **FoundationNetworking** module. Import that module to use it.



**FoundationNetworking**...?


GitHub
Foundation.framework
swift-corelibs-foundation
swift-foundation
swift-foundation-icu
swift-collections

https://github.com/swiftlang/swift-foundation

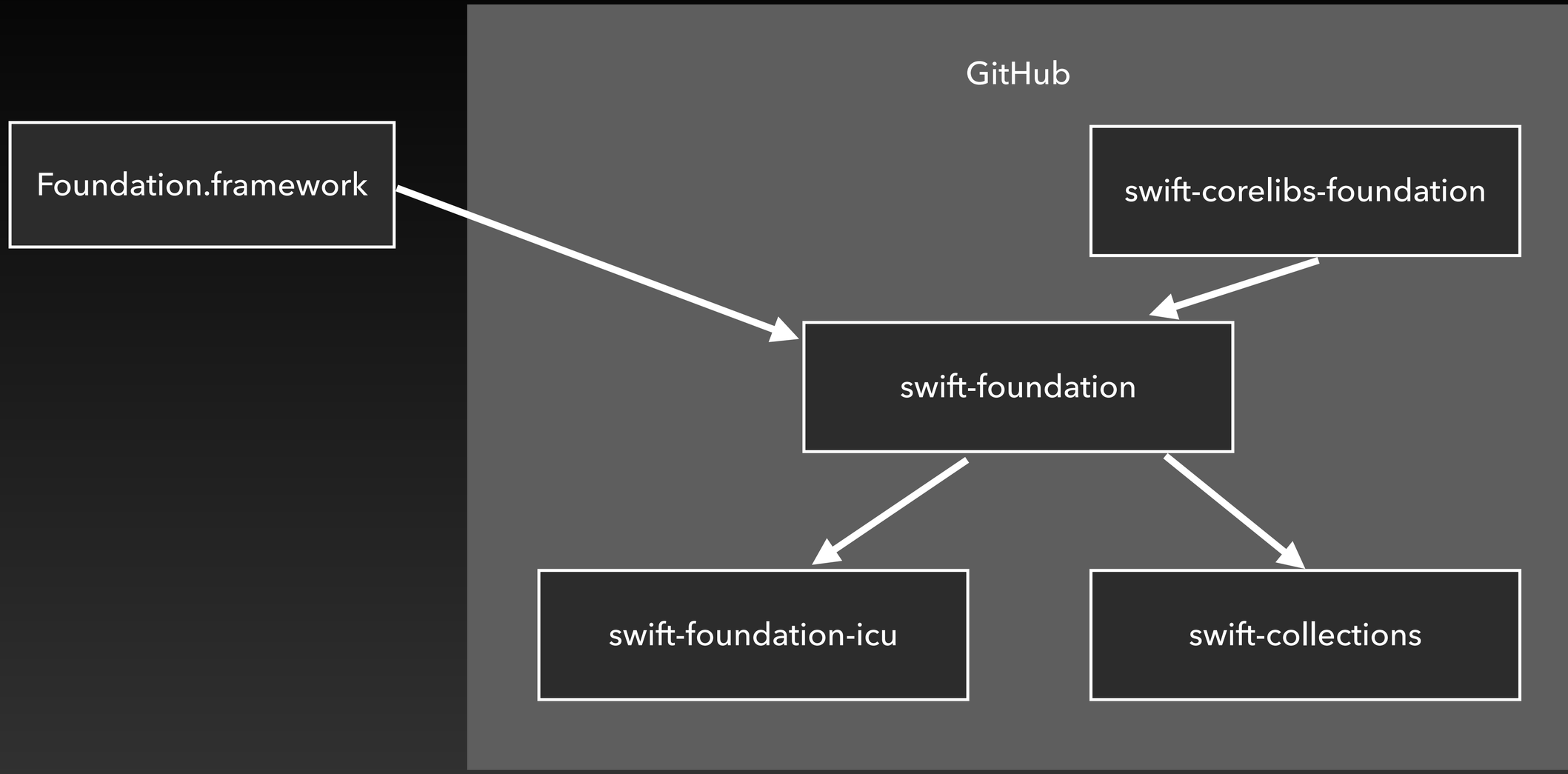
URL

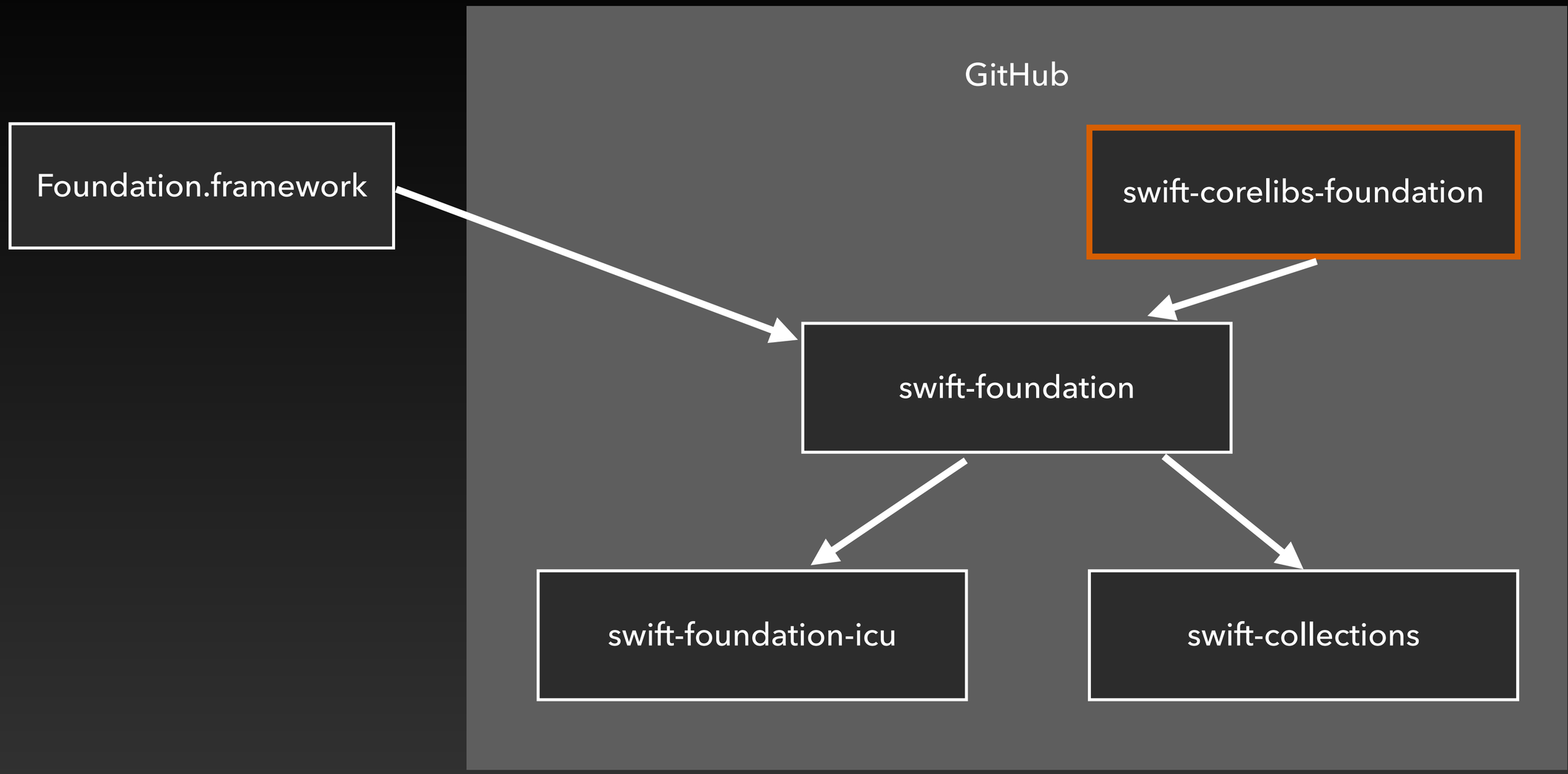
Data

JSONDecoder


Locale

Calendar


GitHub
Foundation.framework
swift-corelibs-foundation
swift-foundation
swift-foundation-icu
swift-collections


GitHub
Foundation.framework
swift-corelibs-foundation
swift-foundation
swift-foundation-icu
swift-collections

https://github.com/swiftlang/swift-corelibs-foundation

NSFormatter

NSKeyedArchiver


URLSession

URLRequest

*swift-corelibs-foundation builds for non-Darwin platforms* ***only***.
*It installs the* `Foundation` *umbrella module,* `FoundationXML`*, and* ***`FoundationNetworking`***.

https://github.com/swiftlang/swift-corelibs-foundation

NSFormatter

NSKeyedArchiver


URLSession

URLRequest

*swift-corelibs-foundation builds for non-Darwin platforms* ***only****.*
*It installs the* `Foundation` *umbrella module,* `FoundationXML`*, and* `FoundationNetworking`*.*

URLSession & Linux?
this is needed

```
import Foundation
#if canImport(FoundationNetworking)
import FoundationNetworking
#endif
```

Everything fixed now?

❌ Error: value of type 'URLSession' has no member 'bytes'

# ❌ Error: value of type 'URLSession' has no member 'bytes'

```
/// Returns a byte stream that conforms to AsyncSequence protocol.
///
/// - Parameter request: The URLRequest for which to load data.
/// - Parameter delegate: Task-specific delegate.
/// - Returns: Data stream and response.
public func bytes(
    for request: URLRequest,
    delegate: (any URLSessionTaskDelegate)? = nil
) async throws -> (
    URLSession.AsyncBytes,
    URLResponse
)
```

# How to solve?

https://github.com/swift-server/async-http-client

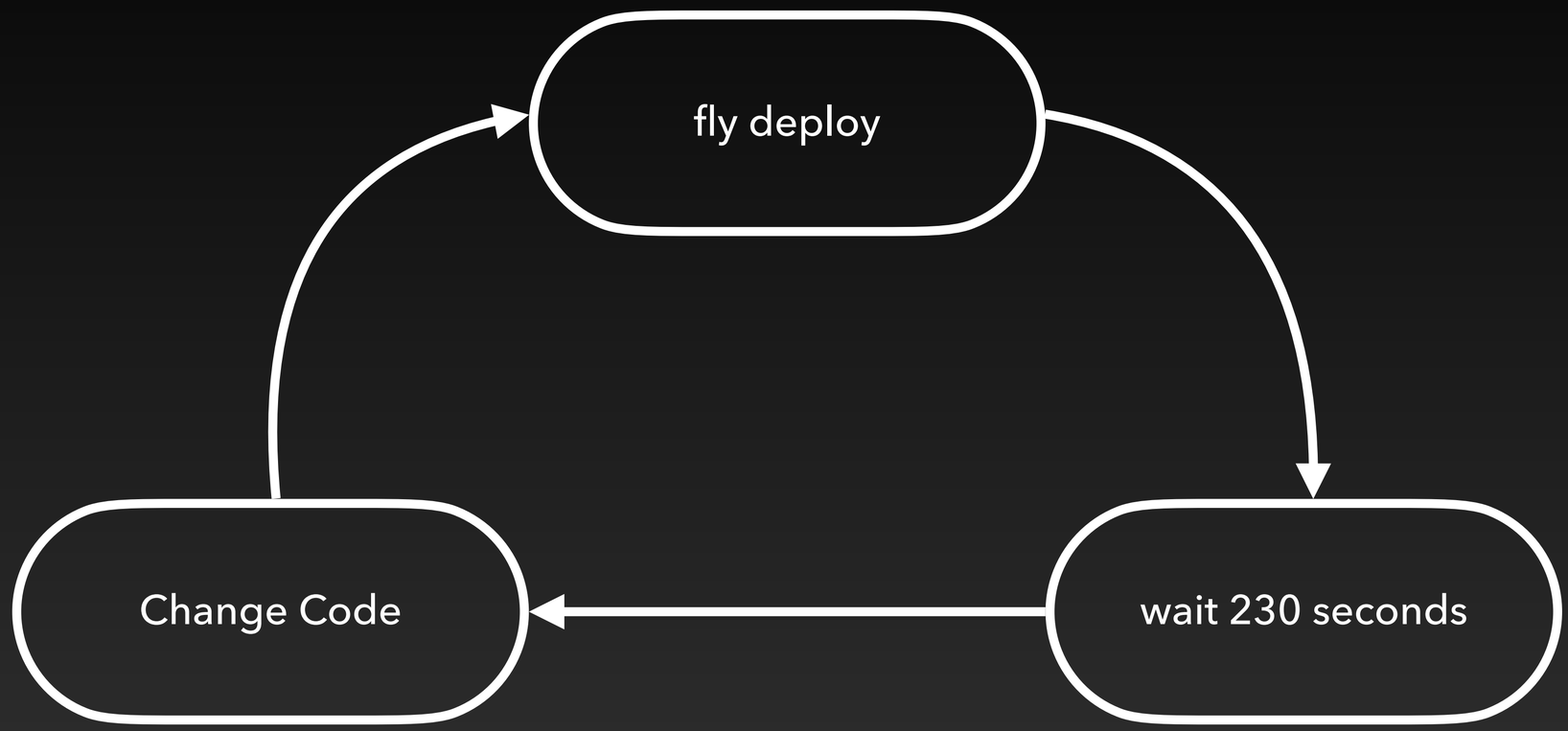
# AsyncHttpClient

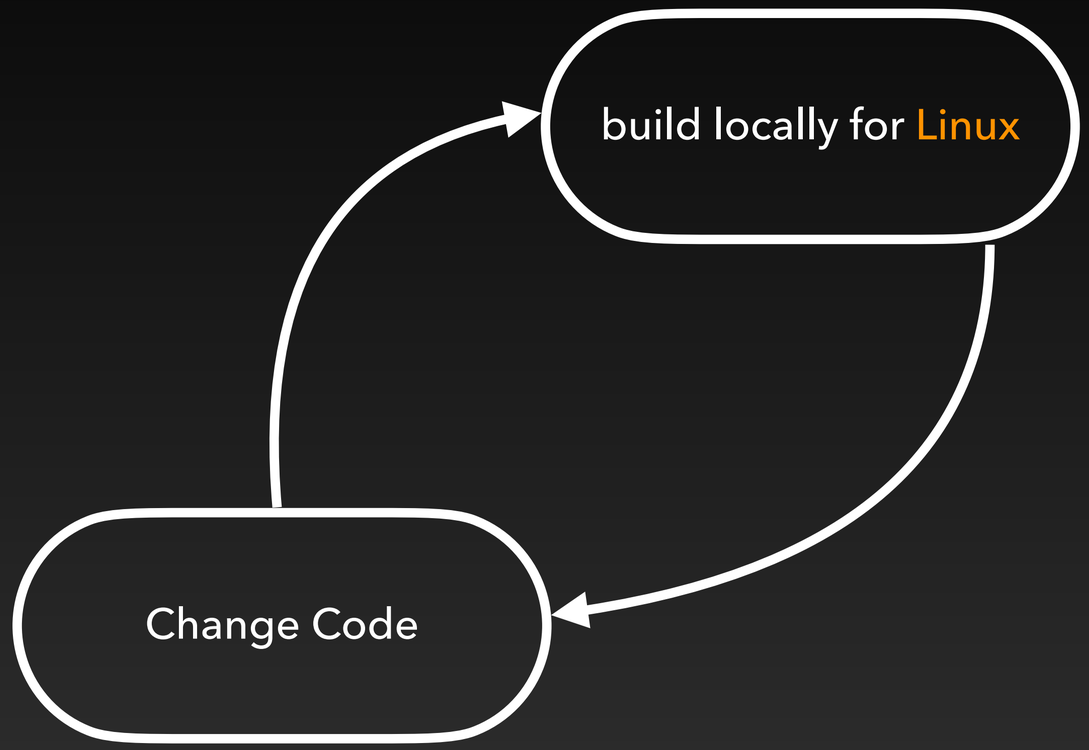
https://github.com/swift-server/async-http-client

# AsyncHttpClient

HTTP Client library built on top of **SwiftNIO**

- First class support for **Swift Concurrency**
- Asynchronous and non-blocking request methods
- Simple follow-redirects (cookie headers are dropped)
- **Streaming body download**
- TLS support
- Automatic HTTP/2 over HTTPS
- Cookie parsing (but not storage)


fly deploy
wait 230 seconds
Change Code


build locally for Linux
Change Code


docker


AtlasBackend — root@255e4873c7d1: /src — docker run --rm -it -v ~/workspace/freiwald/Atlas/AtlasBackend:/src...

```
[~/workspace/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ docker run --rm -it -v "$(pwd):/src" -w "/src" swift     ]
[root@255e4873c7d1:/src# uname -a                                                                                    ]
 Linux 255e4873c7d1 6.6.32-linuxkit #1 SMP Thu Jun 13 14:13:01 UTC 2024 aarch64 aarch64 aarch64 GNU/Linux
[root@255e4873c7d1:/src# swift --version                                                                             ]
 Swift version 6.0.2 (swift-6.0.2-RELEASE)
 Target: aarch64-unknown-linux-gnu
 root@255e4873c7d1:/src#
```

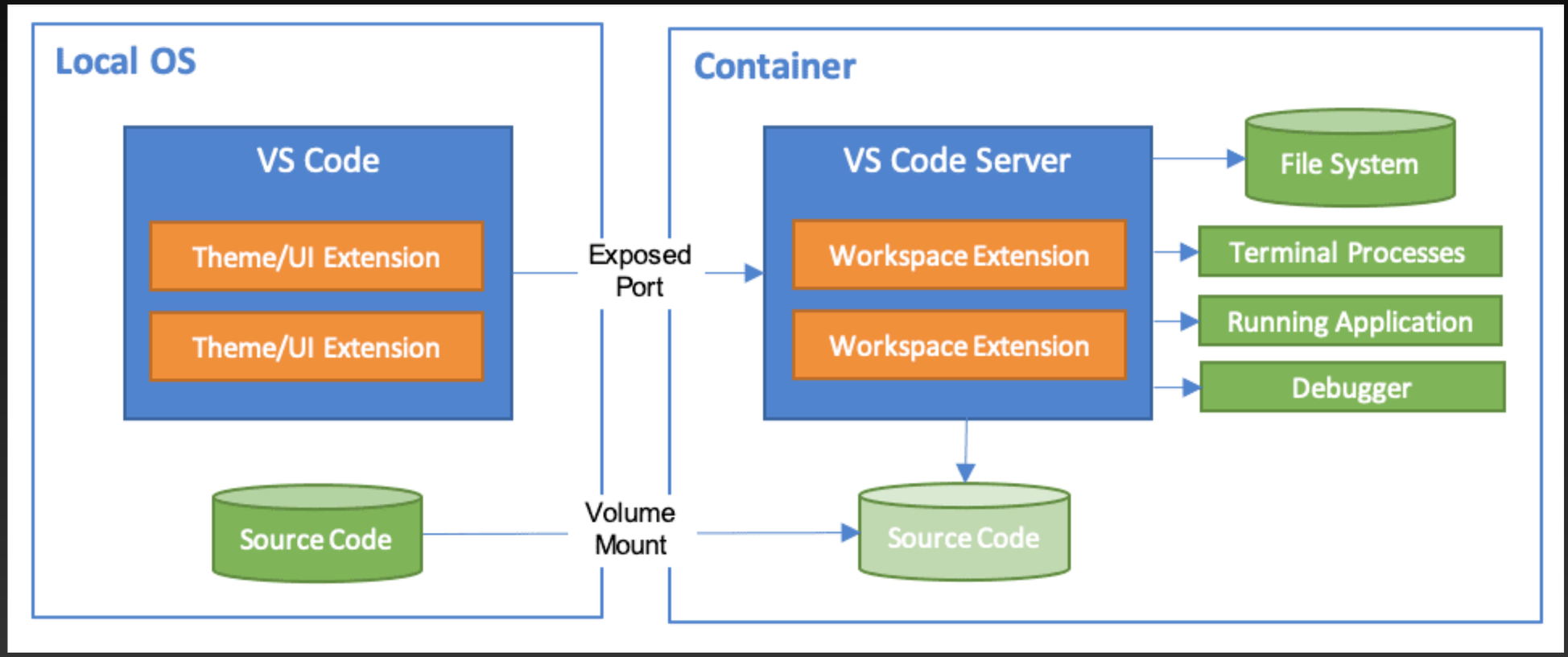

AtlasBackend — root@255e4873c7d1: /src — docker run --rm -it -v ~/workspace/freiwald/Atlas/AtlasBackend:/src...

```
[~/workspace/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ docker run --rm -it -v "$(pwd):/src" -w "/src" swift ]
[root@255e4873c7d1:/src# uname -a                                                                               ]
 Linux 255e4873c7d1 6.6.32-linuxkit #1 SMP Thu Jun 13 14:13:01 UTC 2024 aarch64 aarch64 aarch64 GNU/Linux
[root@255e4873c7d1:/src# swift --version                                                                        ]
 Swift version 6.0.2 (swift-6.0.2-RELEASE)
 Target: aarch64-unknown-linux-gnu
 root@255e4873c7d1:/src#
```


AtlasBackend — root@255e4873c7d1: /src — docker run --rm -it -v ~/workspace/freiwald/Atlas/AtlasBackend:/src...

```
[~/workspace/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ docker run --rm -it -v "$(pwd):/src" -w "/src" swift      ]
root@255e4873c7d1:/src# uname -a                                                                                    ]
Linux 255e4873c7d1 6.6.32-linuxkit #1 SMP Thu Jun 13 14:13:01 UTC 2024 aarch64 aarch64 aarch64 GNU/Linux
root@255e4873c7d1:/src# swift --version                                                                             ]
Swift version 6.0.2 (swift-6.0.2-RELEASE)
Target: aarch64-unknown-linux-gnu
root@255e4873c7d1:/src#
```

# Visual Studio Code Dev Containers



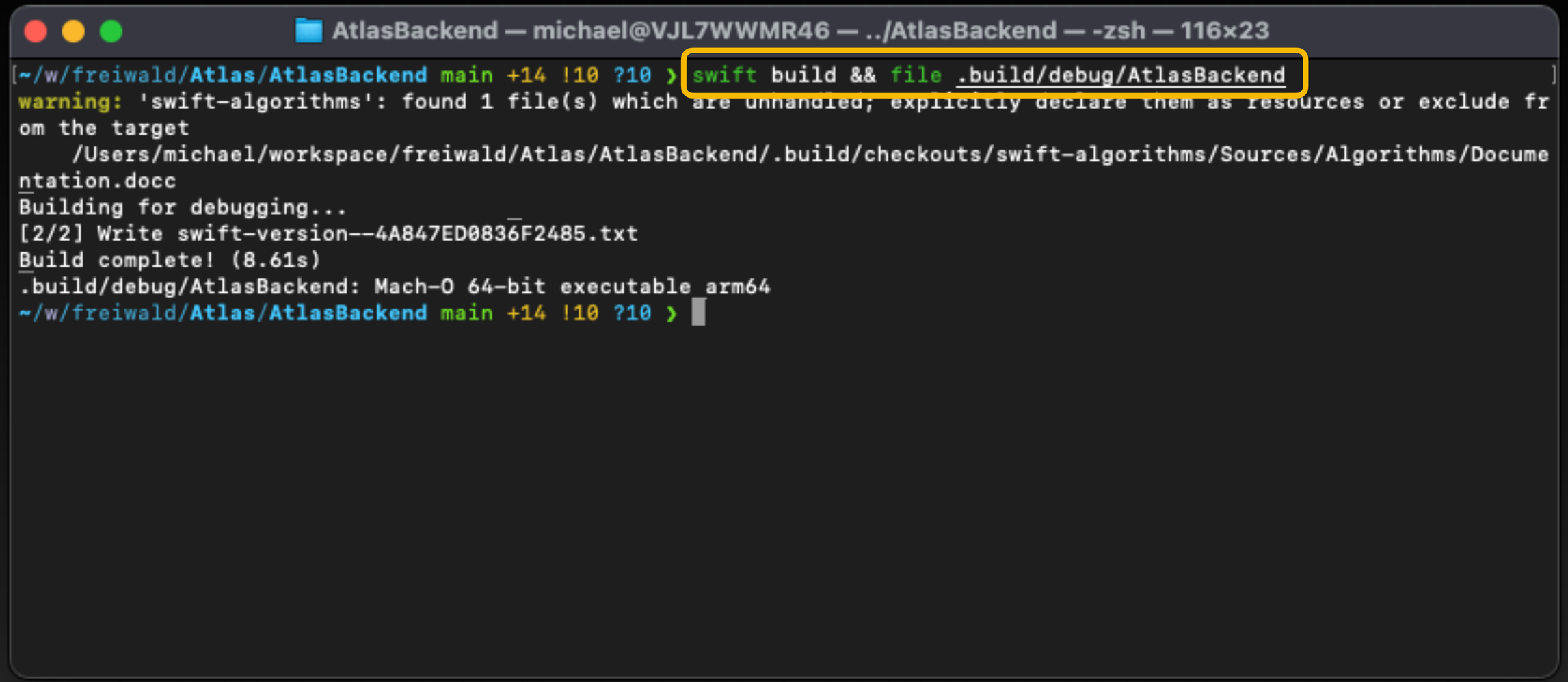
There is a better way

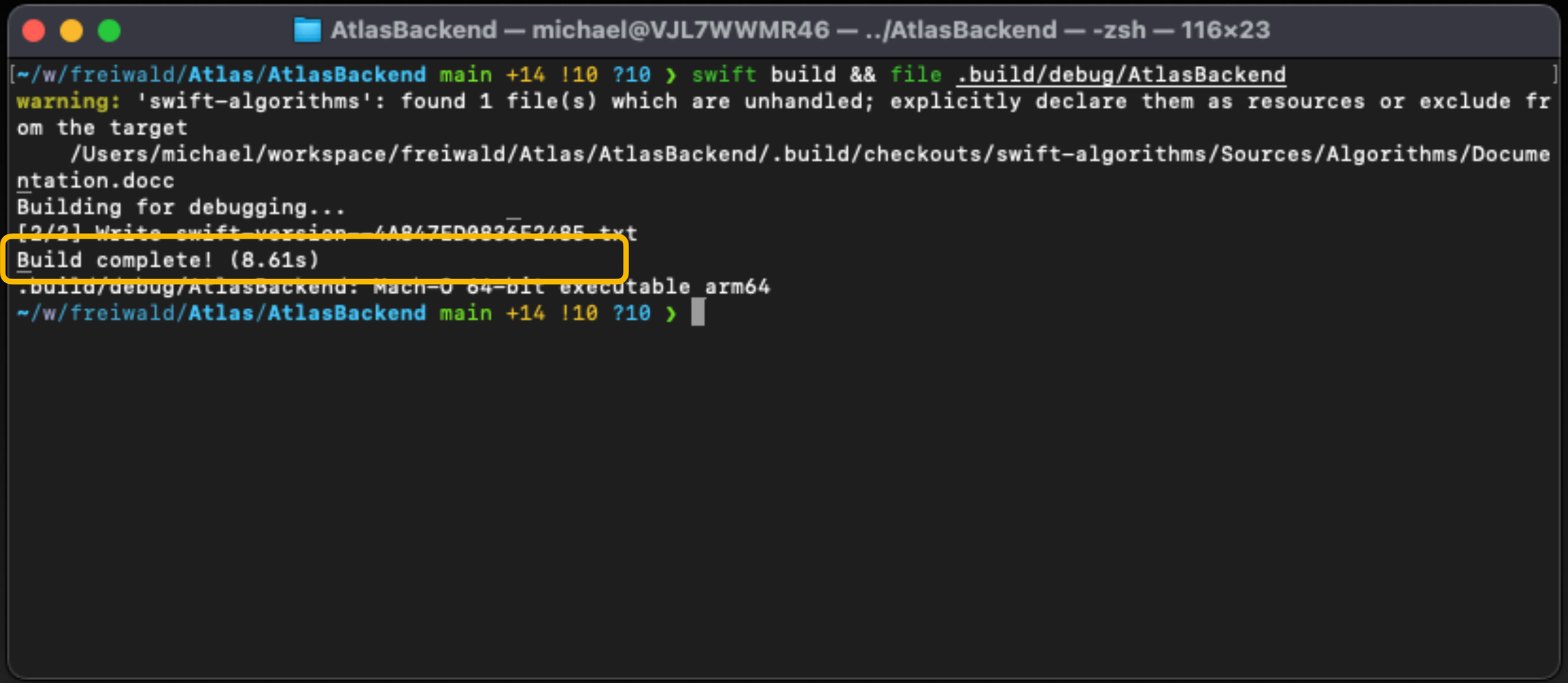
# Swift Static Linux SDK

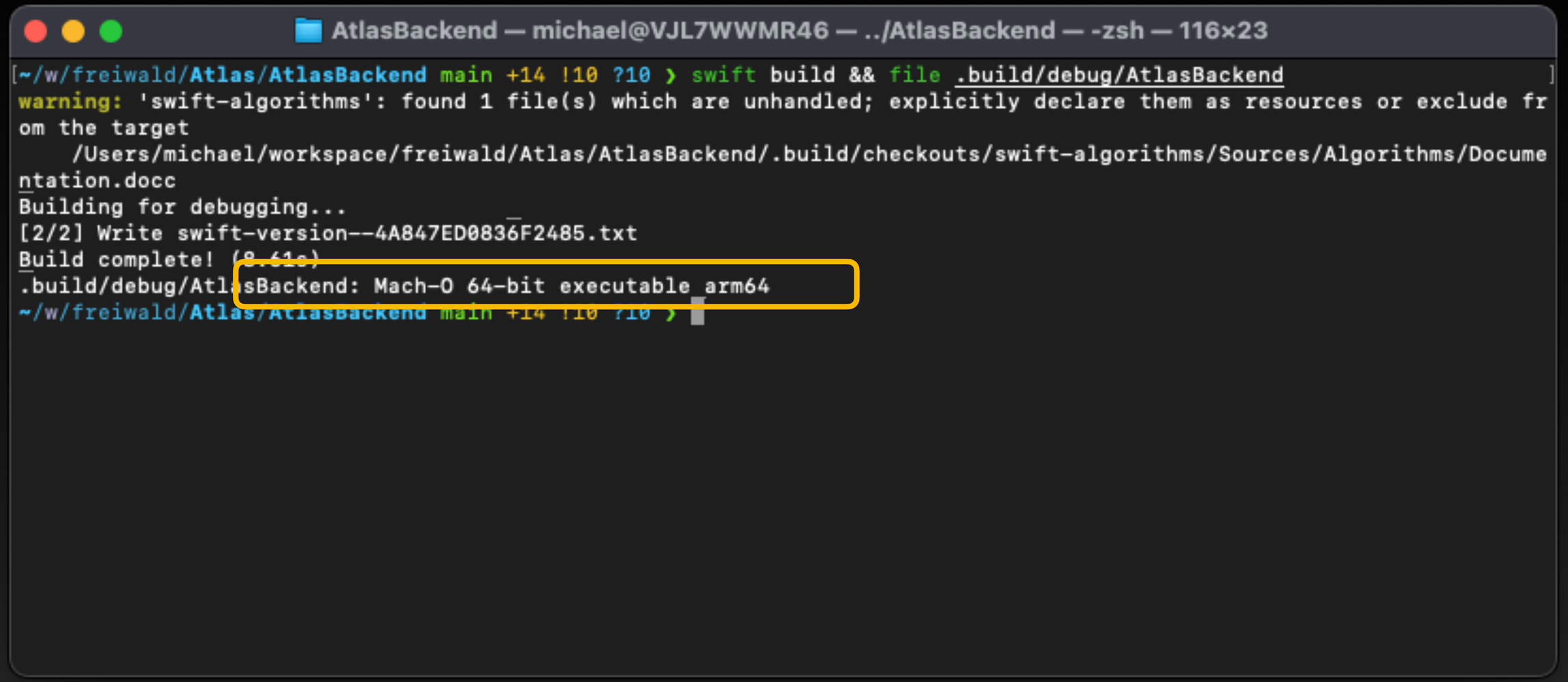
# Swift Static Linux SDK

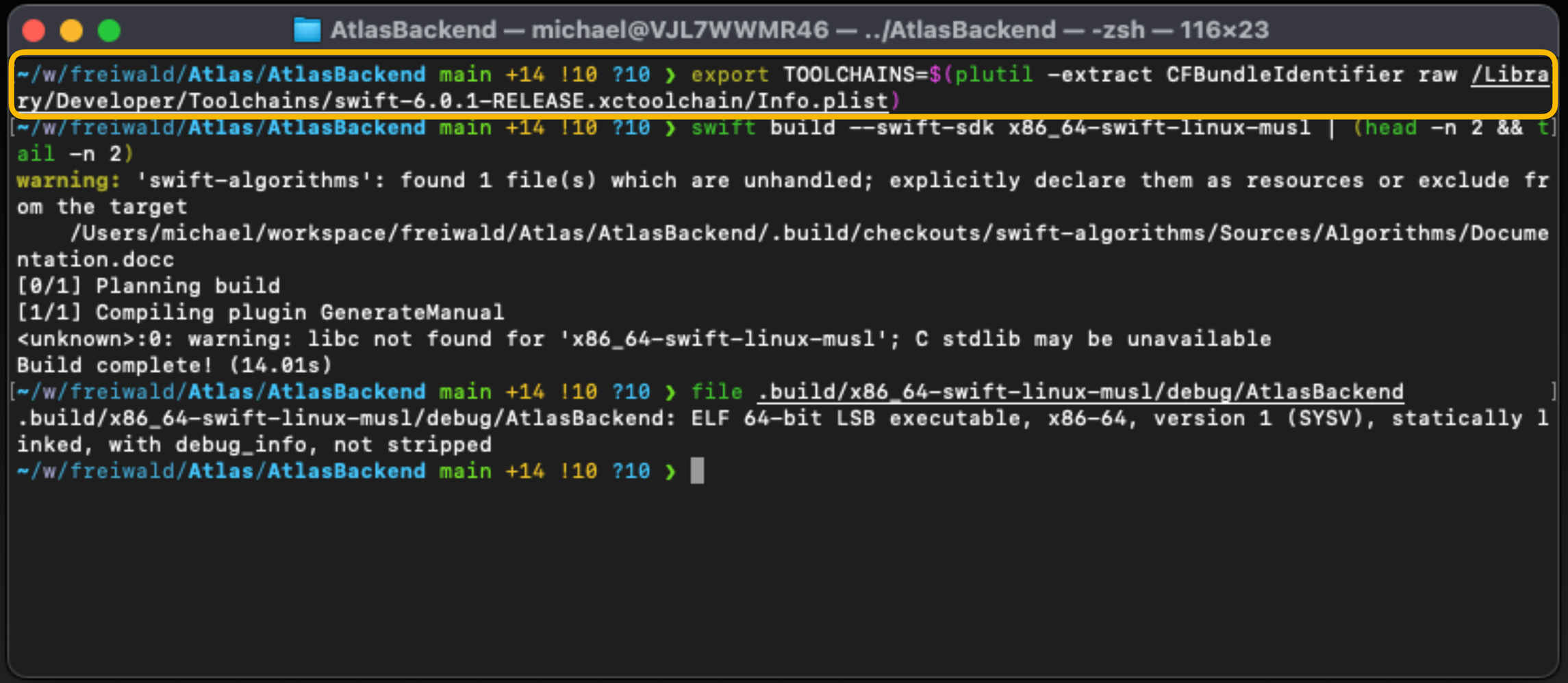
- Install Open Source Toolchain of Swift (Xcode is not enougth)
- Ensure to use this toolchain when building with swift
- Install the Static Linux SDK (same version as the toolchain)

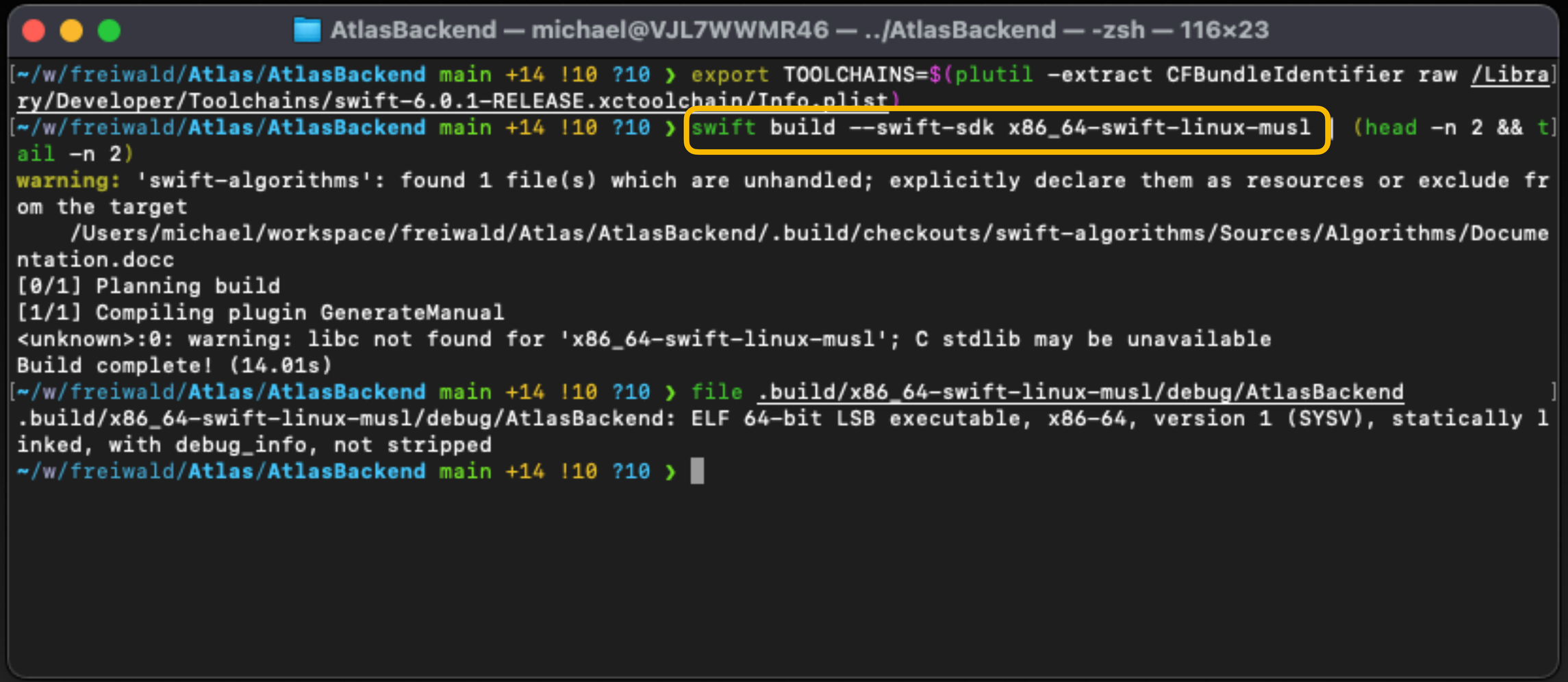
https://www.swift.org/documentation/articles/static-linux-getting-started.html

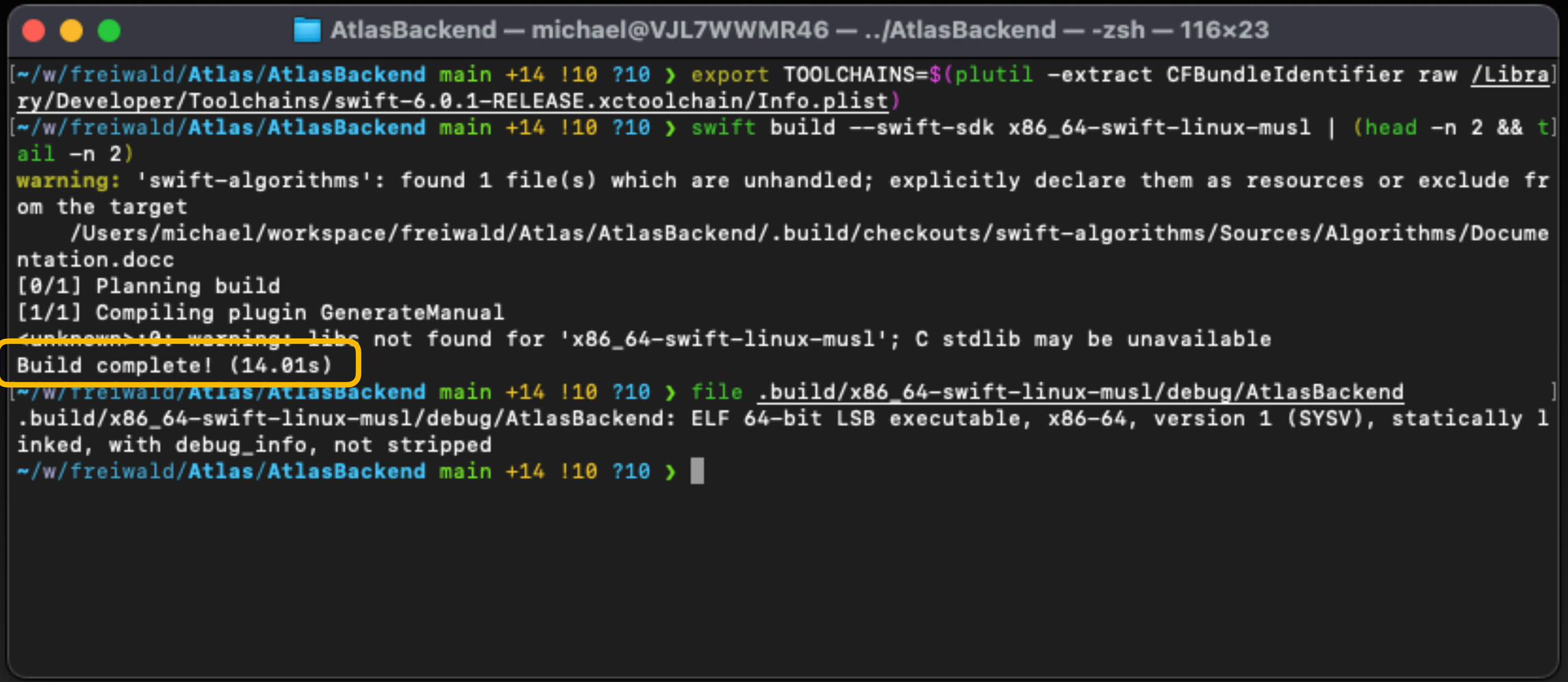

AtlasBackend — michael@VJL7WWMR46 — ../AtlasBackend — -zsh — 116×23
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ swift build && file .build/debug/AtlasBackend
warning: 'swift-algorithms': found 1 file(s) which are unhandled; explicitly declare them as resources or exclude fr
om the target
/Users/michael/workspace/freiwald/Atlas/AtlasBackend/.build/checkouts/swift-algorithms/Sources/Algorithms/Docume
ntation.docc
Building for debugging...
[2/2] Write swift-version--4A847ED0836F2485.txt
Build complete! (8.61s)
.build/debug/AtlasBackend: Mach-O 64-bit executable arm64
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯

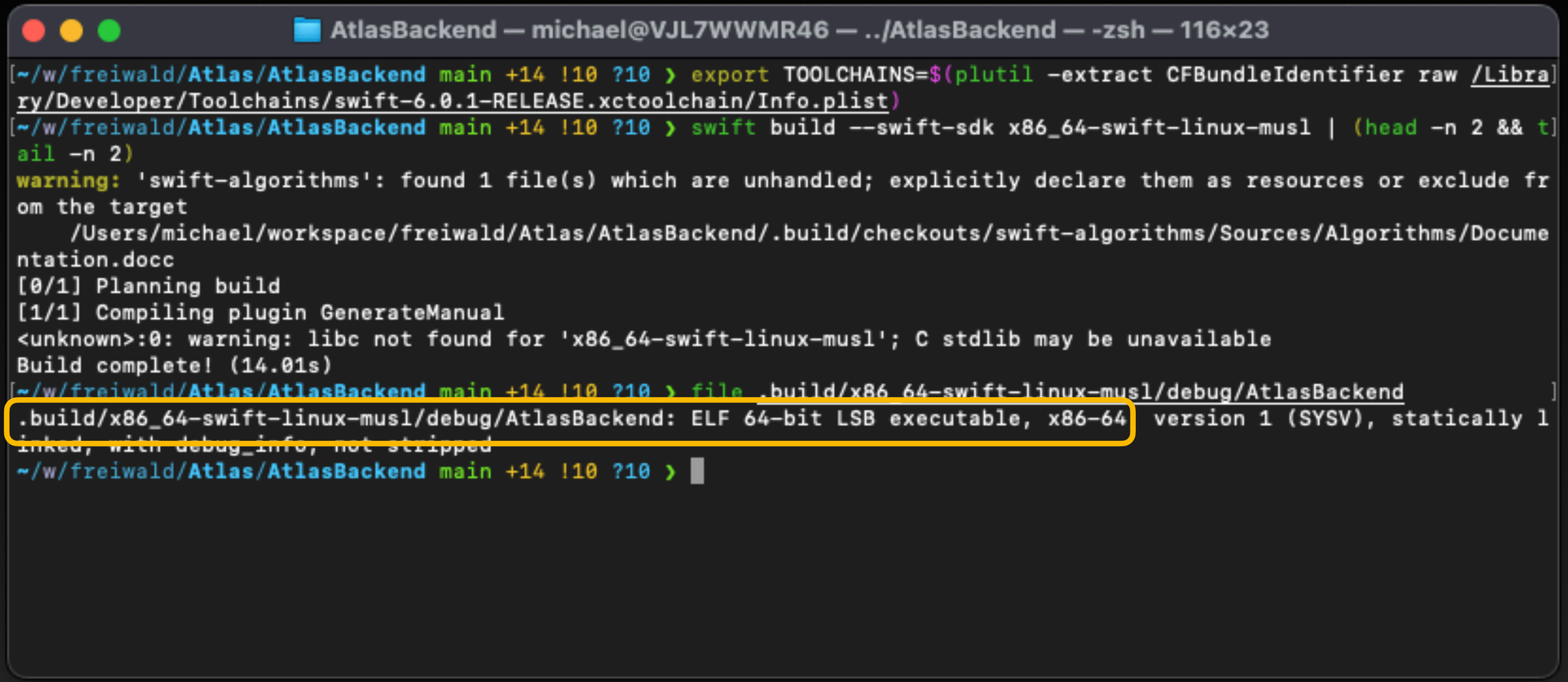

AtlasBackend — michael@VJL7WWMR46 — ../AtlasBackend — -zsh — 116×23
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ swift build && file .build/debug/AtlasBackend
warning: 'swift-algorithms': found 1 file(s) which are unhandled; explicitly declare them as resources or exclude fr
om the target
/Users/michael/workspace/freiwald/Atlas/AtlasBackend/.build/checkouts/swift-algorithms/Sources/Algorithms/Docume
ntation.docc
Building for debugging...
Build complete! (8.61s)
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯


AtlasBackend — michael@VJL7WWMR46 — ../AtlasBackend — -zsh — 116×23
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ swift build && file .build/debug/AtlasBackend
warning: 'swift-algorithms': found 1 file(s) which are unhandled; explicitly declare them as resources or exclude fr
om the target
/Users/michael/workspace/freiwald/Atlas/AtlasBackend/.build/checkouts/swift-algorithms/Sources/Algorithms/Docume
ntation.docc
Building for debugging...
[2/2] Write swift-version--4A847ED0836F2485.txt
Build complete!
.build/debug/AtlasBackend: Mach-O 64-bit executable arm64
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯


AtlasBackend — michael@VJL7WWMR46 — ../AtlasBackend — -zsh — 116×23
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ export TOOLCHAINS=$(plutil -extract CFBundleIdentifier raw /Library/Developer/Toolchains/swift-6.0.1-RELEASE.xctoolchain/Info.plist)
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ swift build --swift-sdk x86_64-swift-linux-musl | (head -n 2 && tail -n 2)
warning: 'swift-algorithms': found 1 file(s) which are unhandled; explicitly declare them as resources or exclude from the target
/Users/michael/workspace/freiwald/Atlas/AtlasBackend/.build/checkouts/swift-algorithms/Sources/Algorithms/Documentation.docc
[0/1] Planning build
[1/1] Compiling plugin GenerateManual
<unknown>:0: warning: libc not found for 'x86_64-swift-linux-musl'; C stdlib may be unavailable
Build complete! (14.01s)
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ file .build/x86_64-swift-linux-musl/debug/AtlasBackend
.build/x86_64-swift-linux-musl/debug/AtlasBackend: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, with debug_info, not stripped
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯


AtlasBackend — michael@VJL7WWMR46 — ../AtlasBackend — -zsh — 116×23
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ export TOOLCHAINS=$(plutil -extract CFBundleIdentifier raw /Library/Developer/Toolchains/swift-6.0.1-RELEASE.xctoolchain/Info.plist)
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ swift build --swift-sdk x86_64-swift-linux-musl | (head -n 2 && tail -n 2)
warning: 'swift-algorithms': found 1 file(s) which are unhandled; explicitly declare them as resources or exclude from the target
/Users/michael/workspace/freiwald/Atlas/AtlasBackend/.build/checkouts/swift-algorithms/Sources/Algorithms/Documentation.docc
[0/1] Planning build
[1/1] Compiling plugin GenerateManual
<unknown>:0: warning: libc not found for 'x86_64-swift-linux-musl'; C stdlib may be unavailable
Build complete! (14.01s)
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ file .build/x86_64-swift-linux-musl/debug/AtlasBackend
.build/x86_64-swift-linux-musl/debug/AtlasBackend: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, with debug_info, not stripped
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯


AtlasBackend — michael@VJL7WWMR46 — ../AtlasBackend — -zsh — 116×23
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ export TOOLCHAINS=$(plutil -extract CFBundleIdentifier raw /Library/Developer/Toolchains/swift-6.0.1-RELEASE.xctoolchain/Info.plist)
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ swift build --swift-sdk x86_64-swift-linux-musl | (head -n 2 && tail -n 2)
warning: 'swift-algorithms': found 1 file(s) which are unhandled; explicitly declare them as resources or exclude from the target
/Users/michael/workspace/freiwald/Atlas/AtlasBackend/.build/checkouts/swift-algorithms/Sources/Algorithms/Documentation.docc
[0/1] Planning build
[1/1] Compiling plugin GenerateManual
not found for 'x86_64-swift-linux-musl'; C stdlib may be unavailable
Build complete! (14.01s)
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ file .build/x86_64-swift-linux-musl/debug/AtlasBackend
.build/x86_64-swift-linux-musl/debug/AtlasBackend: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, with debug_info, not stripped
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯


AtlasBackend — michael@VJL7WWMR46 — ../AtlasBackend — -zsh — 116×23
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ export TOOLCHAINS=$(plutil -extract CFBundleIdentifier raw /Library/Developer/Toolchains/swift-6.0.1-RELEASE.xctoolchain/Info.plist)
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ swift build --swift-sdk x86_64-swift-linux-musl | (head -n 2 && tail -n 2)
warning: 'swift-algorithms': found 1 file(s) which are unhandled; explicitly declare them as resources or exclude from the target
/Users/michael/workspace/freiwald/Atlas/AtlasBackend/.build/checkouts/swift-algorithms/Sources/Algorithms/Documentation.docc
[0/1] Planning build
[1/1] Compiling plugin GenerateManual
<unknown>:0: warning: libc not found for 'x86_64-swift-linux-musl'; C stdlib may be unavailable
Build complete! (14.01s)
[~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯ file .build/x86_64-swift-linux-musl/debug/AtlasBackend ]
.build/x86_64-swift-linux-musl/debug/AtlasBackend: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, with debug_info, not stripped
~/w/freiwald/Atlas/AtlasBackend main +14 !10 ?10 ❯

# Upload to a Linux machine

# Build on Mac
# Run anywhere

# Build on Mac
# Run anywhere

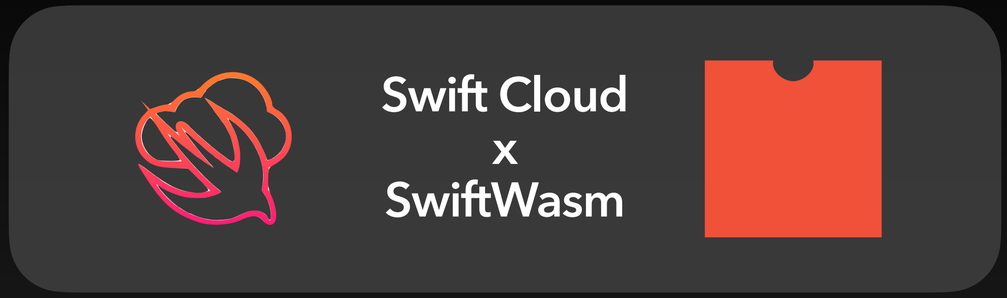
- Xcode can only change Toolchain, no build arguments for SPM
- VSCode for server development
- Use static linux sdk to check compability

# Build on Mac
# Run anywhere

- Build Linux binary on macOS
- Uploaded binary to a Linux machine
- Run app on Linux
- Make a request

# Summary

- Keep our API-Keys out of the client app
- Swift on Server works
- Be careful when selection 3rd-party-SDKs
- check Linux compabability if needed
- This is just the beginning





# One more thing

How to secure your own Backend?

# One more thing

- API-Keys (on client-side 🤪)
- Certificate Pinning (with your own server certificate)
- App Attest (to ensure that requests your server receives come from legitimate instances of your app)
- Validate app receipes from App Store
- Hope for Apple Intelligence SDK 🙏

# Thank you

https://freiwald.dev