

HONZA DVORSKY

@czechboy0

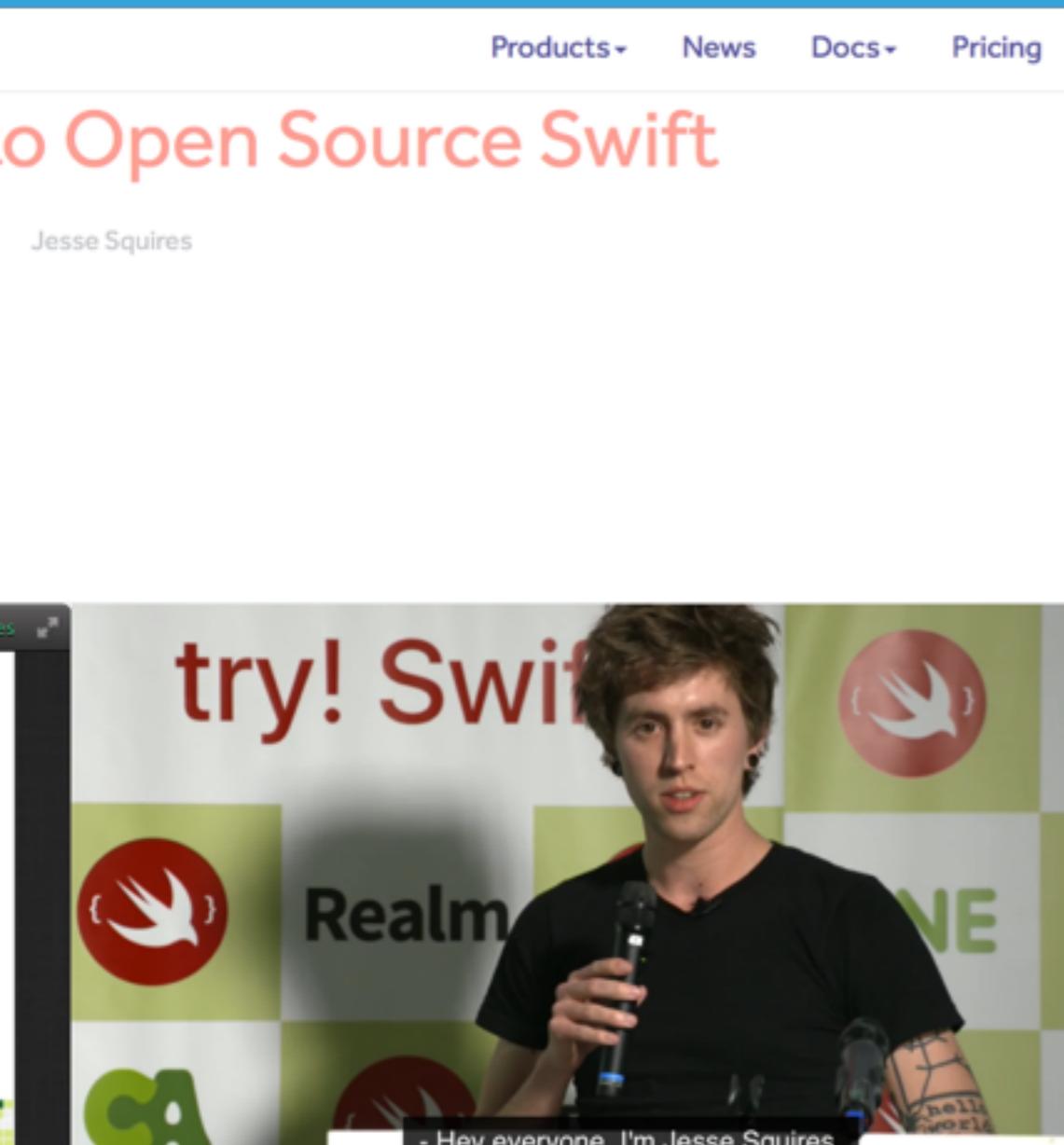
honzadvorsky.com

NSLondon 30.06.2016

SWIFT PACKAGE MANAGER

Swift Package Manager

SwiftPM



CONTRIBUTING TO OPEN SOURCE SWIFT

<https://realm.io/news/tryswift-jesse-squires-contributing-open-source-swift/>

AGENDA

- ▶ introduction to SwiftPM
- ▶ creating a simple package
- ▶ advanced topics

[SwiftPM] is a tool for managing the distribution of Swift code.
It's integrated with the Swift build system to automate the process of downloading, compiling and linking dependencies.

swift.org/package-manager

SwiftPM is a

- ▶ dependency manager
- ▶ build tool
- ▶ test tool

SwiftPM is

- ▶ command line based
- ▶ cross-platform (macOS, Linux)
- ▶ decentralized
- ▶ opinionated
- ▶ convention over configuration

Where to learn more about it

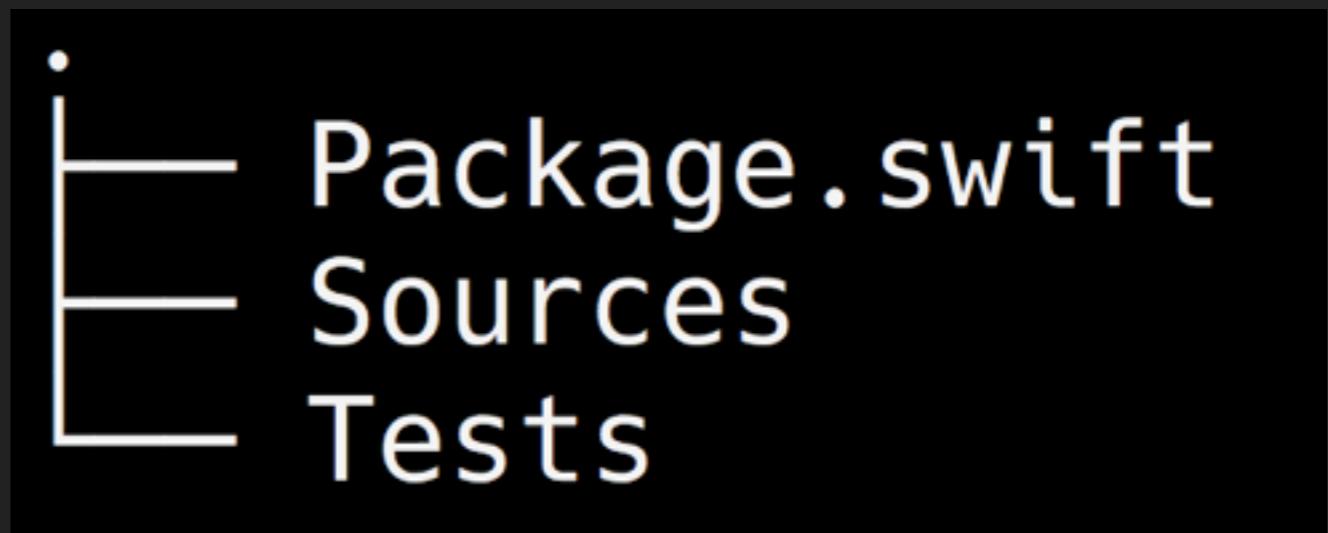
- ▶ swift.org/package-manager
- ▶ github.com/apple/swift-package-manager
- ▶ Mailing list: [swift-build-dev](mailto:swift-build-dev@swift.org)
- ▶ Slack: <https://swift-package-manager.herokuapp.com>

Swift Package Manager

Swift Package Manager

Package

- ▶ is a folder
- ▶ Package.swift
- ▶ source files



Example package: Environment

- ▶ 1 library module
- ▶ 1 test module

```
[→ Environment git:(master) ✘ tree
```

```
.
```

```
└── Package.swift
```

```
└── Sources
```

```
    └── Environment
```

```
        └── Environment.swift
```

```
└── Tests
```

```
    └── Environment
```

```
        └── EnvironmentTests.swift
```

```
└── LinuxMain.swift
```

```
> Package.swift ✘
```

```
import PackageDescription
```



```
let package = Package(
```

```
    name: "Environment"
```

```
)
```

“I ALREADY USE COCOAPODS/
CARTHAGE, IS THIS JUST ANOTHER
DEPENDENCY MANAGER?”

you

“I ALREADY SUPPORT COCOAPODS,
HOW CAN I SUPPORT SWIFTPM?”

you

CocoaPods -> SwiftPM

- ▶ <https://github.com/neonichu/schoutedenapus>
- ▶ converts CocoaPods Spec to Package.swift
- ▶ helpful starting point for existing projects

“WHY SHOULD I SUPPORT SWIFTPM?”

you

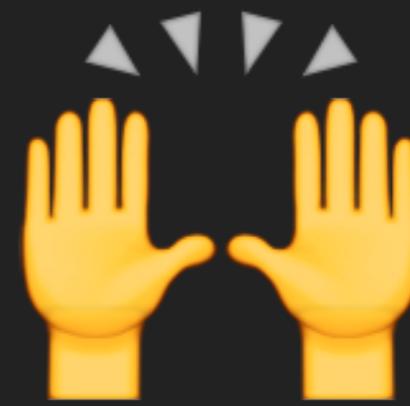
SwiftPM adopters

- ▶ >2000 SwiftPM packages on GitHub
- ▶ stats: github.com/czechboy0/swiftpm-packages-statistics
- ▶ server-side Swift community
- ▶ models, parsers, networking
- ▶ Foundation is available

Installing SwiftPM

- ▶ part of Swift 3 toolchains
 - ▶ swift.org/download
- ▶ Xcode 8
- ▶ swiftenv: github.com/kylef/swiftenv

**GETTING OUR
DIRTY**



Testing (e.g. Travis)

- ▶ no need to run Linux locally
- ▶ scales to more platforms in the future (Linux flavors, Windows, Android)
- ▶ free for OSS projects
- ▶ example: Jay

```
os:  
  - linux  
  - osx  
language: generic  
sudo: required  
dist: trusty  
osx_image: xcode7.3  
install:  
  - eval "$(curl -sL https://  
script:  
  - swift build  
  - swift test  
notifications:  
email:  
  on_success: never  
  on_failure: change
```

Package discovery

- ▶ established: CocoaPods Search
- ▶ new:
 - ▶ Swift Package Catalog: swiftpkgs.ng.bluemix.net
 - ▶ Swift Modules: swiftmodules.com
 - ▶ Twitter, GitHub

Server-side Swift community

- ▶ Zewo: zewo.io
- ▶ Qutheory: qutheory.io
- ▶ Open Swift: github.com/open-swift
- ▶ IBM: github.com/IBM-Swift
- ▶ many more

ADVANCED



Languages supported

- ▶ Swift
- ▶ C
- ▶ Objective-C
- ▶ C++
- ▶ Objective-C++

SwiftPM Commands

- ▶ swift build
- ▶ swift test
- ▶ swift package

swift build

```
[→ ~ swift build --help
```

OVERVIEW: Build sources into binary products

USAGE: swift build [mode] [options]

MODES:

| | |
|-----------------------------|------------------------------------------|
| -c, --configuration <value> | Build with configuration (debug release) |
| --clean <mode> | Delete artifacts (build dist) |

OPTIONS:

| | |
|---------------------|-----------------------------------------------------|
| -C, --chdir <path> | Change working directory before any other operation |
| --build-path <path> | Specify build directory |
| --color <mode> | Specify color mode (auto always never) |
| -v, --verbose | Increase verbosity of informational output |
| -Xcc <flag> | Pass flag through to all C compiler invocations |
| -Xlinker <flag> | Pass flag through to all linker invocations |
| -Xswiftc <flag> | Pass flag through to all Swift compiler invocations |

NOTE: Use `swift package` to perform other functions on packages

```
[→ ~ ]
```

swift test

```
[→ ~ swift test --help
```

OVERVIEW: Build and run tests

USAGE: swift test [specifier] [options]

SPECIFIER:

| | |
|--------------------------------------------------|----------------------------|
| -s, --specifier <test-module>.<test-case> | Run a test case subclass |
| -s, --specifier <test-module>.<test-case>/<test> | Run a specific test method |

OPTIONS:

| | |
|---------------------|-----------------------------------------------------|
| -C, --chdir <path> | Change working directory before any other operation |
| --build-path <path> | Specify build directory |

NOTE: Use `swift package` to perform other functions on packages

swift package

[→ ~ swift package

OVERVIEW: Perform operations on Swift packages

USAGE: swift package [command] [options]

COMMANDS:

| | |
|---------------------------------------|-----------------------------------------|
| init [--type <type>] | Initialize package (library executable) |
| fetch | Fetch package dependencies |
| update | Update package dependencies |
| generate-xcodeproj [--output <path>] | Generates an Xcode project |
| show-dependencies [--format <format>] | Print dependency graph (text dot json) |
| dump-package [--output <path>] | Print Package.swift as JSON |

OPTIONS:

| | |
|--------------------|-----------------------------------------------------|
| -C, --chdir <path> | Change working directory before any other operation |
| --color <mode> | Specify color mode (auto always never) |
| -v, --verbose | Increase verbosity of informational output |
| --version | Print the Swift Package Manager version |
| -Xcc <flag> | Pass flag through to all C compiler invocations |
| -Xlinker <flag> | Pass flag through to all linker invocations |
| -Xswiftc <flag> | Pass flag through to all Swift compiler invocations |

NOTE: Use `swift build` to build packages, and `swift test` to test packages

Package

- ▶ collection of modules (targets)
- ▶ module is a collection of sources
- ▶ module types
 - ▶ library (source files, no main.swift)
 - ▶ executable (source files, 1 main.swift)
 - ▶ system module

System module

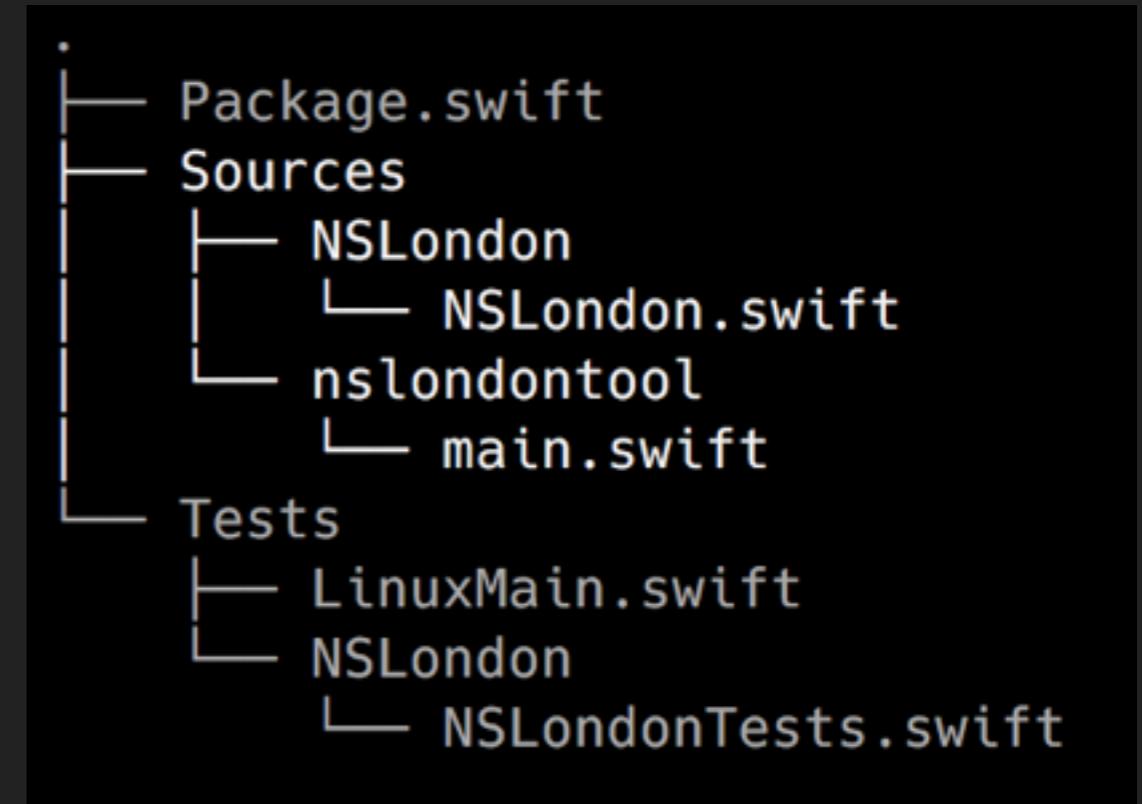
- ▶ using system libraries (e.g. openssl, curl) in SwiftPM
- ▶ modulemap
- ▶ how to use:
 - ▶ <http://ankit.im/swift/2016/04/06/compiling-and-interpolating-C-using-swift-package-manager/>
 - ▶ <http://ankit.im/swift/2016/05/21/creating-objc-cpp-packages-with-swift-package-manager/>

Modules

implicit

explicit

```
let package = Package(  
    name: "NSLondon",  
    targets: [  
        Target(name: "NSLondon"),  
        Target(name: "nslondontool", dependencies: ["NSLondon"])  
    ],  
    dependencies: [  
        .Package(url: "https://github.com/czechboy0/Environment.git", m  
    ]  
)
```



Dependencies

```
Package.swift      x

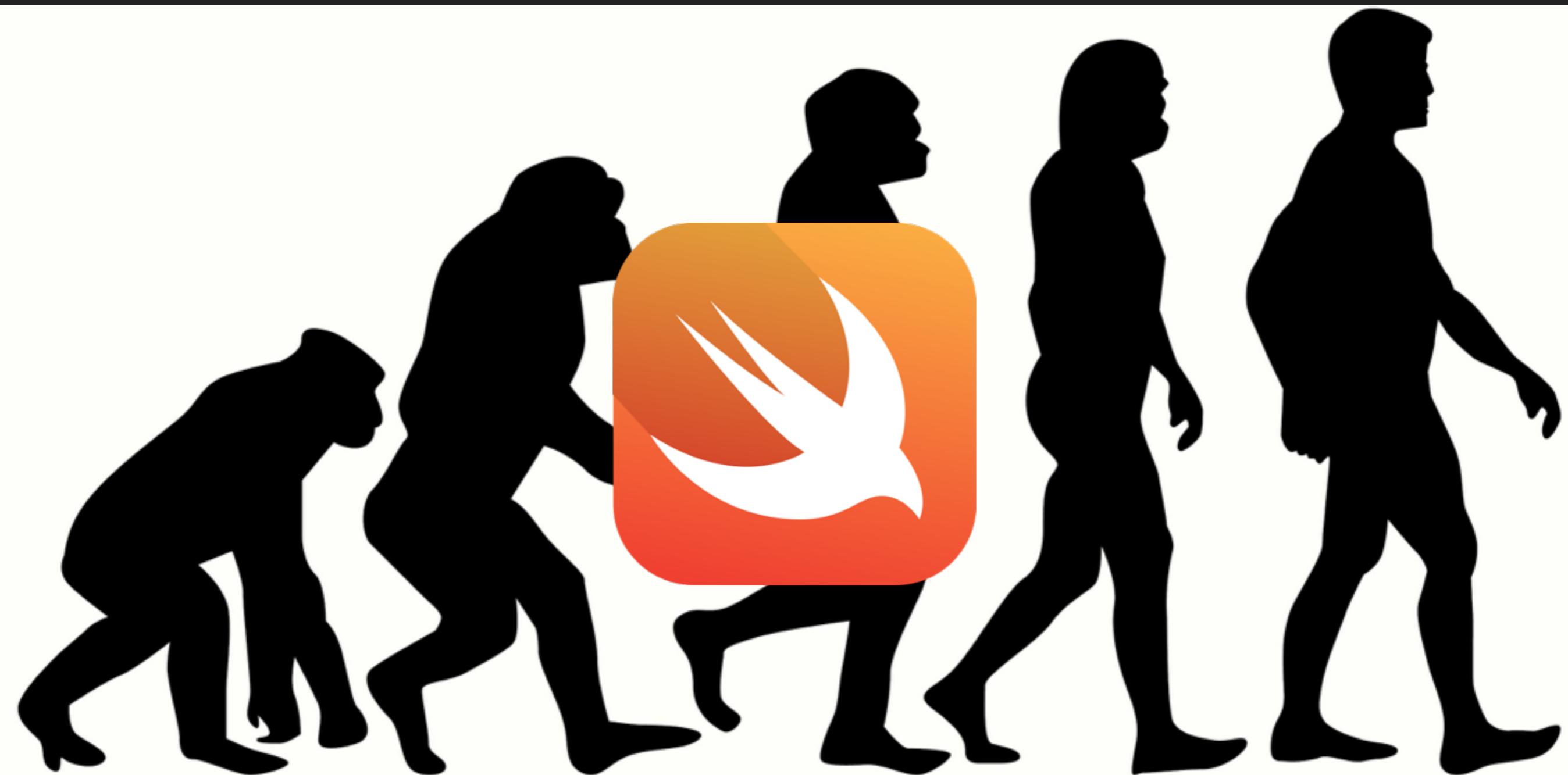
import PackageDescription

let package = Package(
    name: "NSLondon",
    targets: [
        Target(name: "NSLondon"),
        Target(name: "nslondontool", dependencies: ["NSLondon"])
    ],
    dependencies: [
        .Package(url: "https://github.com/czechboy0/Environment.git", majorVersion: 0)
    ]
)
```

swift package show-dependencies

```
[→ swift-package-crawler git:(master) ✘ swift package show-dependencies

.
└ HTTPSCClient<https://github.com/VeniceX/HTTPSCClient.git@0.8.0>
    └ TCPSSL<https://github.com/VeniceX/TCPSSL.git@0.8.0>
        └ TCP<https://github.com/VeniceX/TCP.git@0.8.0>
            └ IP<https://github.com/VeniceX/IP.git@0.8.0>
                └ Venice<https://github.com/VeniceX/Venice.git@0.8.0>
                    └ CLibvenice<https://github.com/VeniceX/CLibvenice.git@0.5.0>
                    └ C7<https://github.com/open-swift/C7.git@0.8.1>
                    └ POSIX<https://github.com/Zewo/POSIX.git@0.5.0>
            └ OpenSSL<https://github.com/Zewo/OpenSSL.git@0.8.1>
                └ File<https://github.com/VeniceX/File.git@0.8.0>
                    └ Venice<https://github.com/VeniceX/Venice.git@0.8.0>
                    └ CLibvenice<https://github.com/VeniceX/CLibvenice.git@0.5.0>
                    └ C7<https://github.com/open-swift/C7.git@0.8.1>
                    └ String<https://github.com/Zewo/String.git@0.7.2>
                    └ POSIX<https://github.com/Zewo/POSIX.git@0.5.0>
                └ COpenSSL<https://github.com/Zewo/COpenSSL.git@0.6.0>
            └ HTTPParser<https://github.com/Zewo/HTTPParser.git@0.9.0>
                └ CHTTPPParser<https://github.com/Zewo/CHTPPParser.git@0.5.0>
                └ URI<https://github.com/Zewo/URI.git@0.8.0>
                    └ String<https://github.com/Zewo/String.git@0.7.2>
                    └ CURIParser<https://github.com/Zewo/CURIParser.git@0.5.0>
                    └ C7<https://github.com/open-swift/C7.git@0.8.1>
                └ S4<https://github.com/open-swift/S4.git@0.9.0>
```



SwiftPM Evolution

- ▶ Community Proposal
- ▶ SE-0019 Package Manager Testing
- ▶ SE-0038 C Language Target Support
- ▶ SE-0063 System Module Search Paths
- ▶ SE-0082 Editable Packages
- ▶ SE-0085 Command Names

Missing features

- ▶ Xcode integration
- ▶ better iOS/watchOS/tvOS support
- ▶ “you tell me”

Getting Involved ❤

- ▶ start reading the mailing list
- ▶ read documentation, source code
- ▶ join the Slack
- ▶ follow people on Twitter and GitHub

- ▶ Honza Dvorsky 
- ▶ @czechboy0
- ▶ honzadvorsky.com
- ▶ slides:
 - ▶ https://honzadvorsky.com/articles/2016-06-30-19-00-nslondon_swift_package_manager/