



HONZA DVORSKY

@czechboy0

honzadvorsky.com



Swift

- created by Apple
- announced in June 2014
- open sourced December 2015
- Swift 3.0 - September 2016

M
DEV
TALK

Swift Package Manager

SwiftPM

Listen carefully if you're...

- an iOS/macOS developer
- backend developer
- an Android developer
- interested in what Apple sees as the future of programming

Agenda

- introduction to SwiftPM
- demo
- 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
- Slack: <https://swift-package-manager.herokuapp.com>

M
DEV
TALK

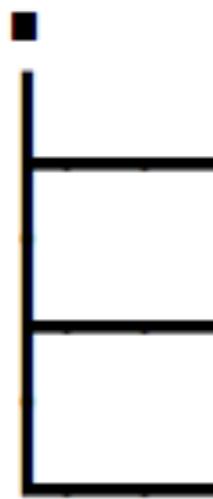
Swift Package Manager

M
DEV
TALK

Swift Package Manager

Package

- is a folder
- Package.swift
- source files
- contains modules



Package.swift
Sources
Tests

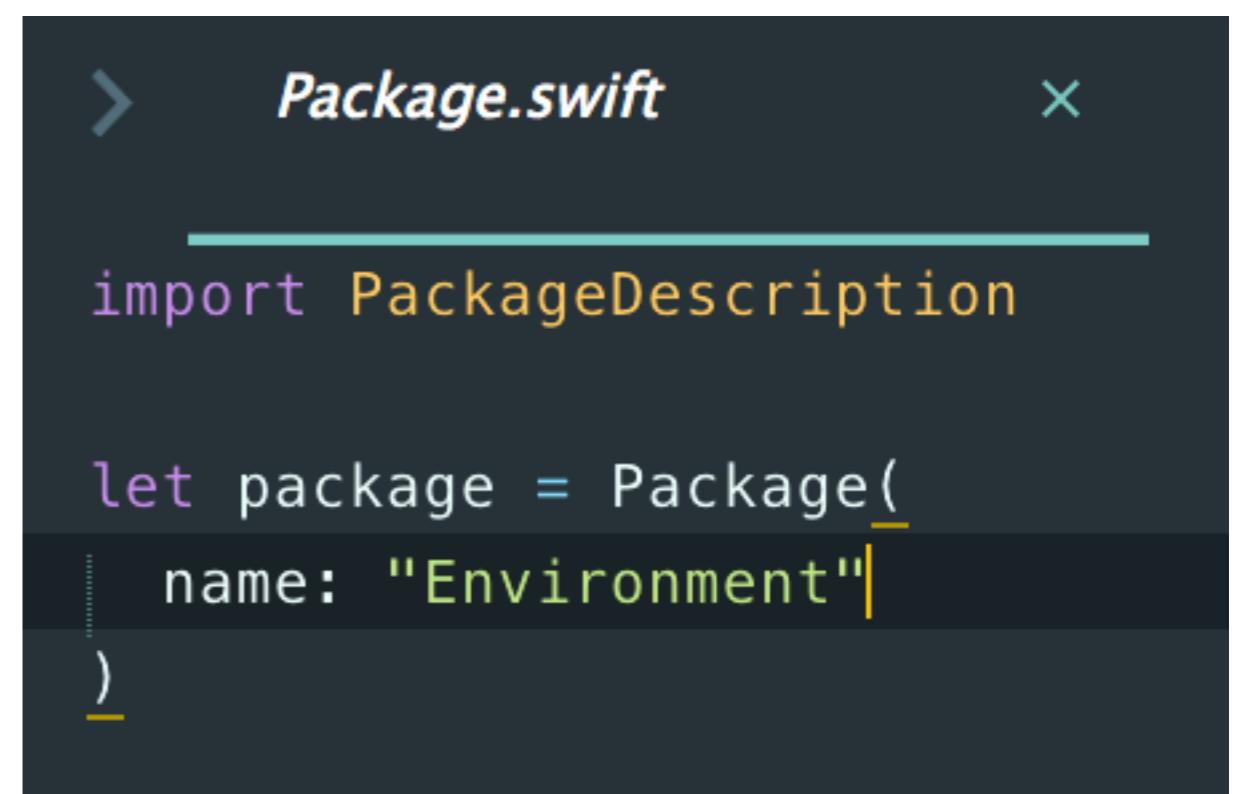
Module

- collection of source files (e.g. .swift files)
- build instructions (e.g. “build as a library”)
- e.g. Foundation, XCTest, Dispatch, ...

Example Package: Environment

- 1 library module
- 1 test module

```
└── Package.swift
└── Sources
    └── Environment
        └── Environment.swift
└── Tests
    └── Environment
        └── EnvironmentTests.swift
└── LinuxMain.swift
```



A screenshot of a code editor showing the `Package.swift` file. The file contains the following Swift code:

```
> 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

- >3500 SwiftPM packages on GitHub
- stats: github.com/czechboy0/swiftpm-packages-statistics
- server-side Swift community
- models, parsers, networking
- Foundation, XCTest, Dispatch available on macOS, Ubuntu

Swift on Android



- [apple/swift port to Android Pull Request](#)
- [Getting Started with Swift on Android](#)
- [How we put an app in the Android Play Store using Swift](#)
- [Google considering Swift for Android](#)

Installing SwiftPM

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

M
DEV
TALK

Getting our  dirty

Demo

1. mDevTool

1.1.extension on String to prepend "mDev" ("Talk" ->
"mDevTalk")

1.2.command line tool to use (1)

2. mDevServer

2.1.Swift server to generate HTML template using (1)

M
DEV
TALK



To Xcode!

M
DEV
TALK



Back To Slides!

Testing with Travis CI

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

```
1  os:
2    - linux
3    - osx
4  language: generic
5  sudo: required
6  dist: trusty
7  osx_image: xcode7.3
8  install:
9    - eval "$(curl -sL https://gist.github.com/...)"
10 script:
11   - swift build
12   - swift test
13 notifications:
14   email:
15     on_success: never
16     on_failure: change
```

.travis.yml in Environment

Package discovery

- established:
 - CocoaPods Search
- new:
 - Swift Package Catalog: swiftpkgs.ng.bluemix.net
 - libraries.io
 - Twitter, GitHub

Server-side Swift community

- Vapor: vapor.codes
- IBM: github.com/IBM-Swift
- Zewo: zewo.io
- Open Swift: github.com/open-swift
- many more

M
DEV
TALK

Advanced



M
DEV
TALK

Swift Package Manager

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) [default: debug]
--clean [<mode>]	Delete artifacts (build dist) [default: build]

OPTIONS:

-C, --chdir <path>	Change working directory before any other operation
--build-path <path>	Specify build directory [default: ./build]
--color <mode>	Specify color mode (auto always never) [default: auto]
-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 [options]

OPTIONS:

-s, --specifier <test-module>.<test-case>	Run a test case subclass
-s, --specifier <test-module>.<test-case>/<test>	Run a specific test method
-l, --list-tests	Lists test methods in specifier format
-C, --chdir <path>	Change working directory before any other operation
--build-path <path>	Specify build directory
--color <mode>	Specify color mode (auto always never) [default: auto]
-v, --verbose	Increase verbosity of informational output
--skip-build	Skip building the test target
-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 package

```
[→ ~ swift package --help
```

OVERVIEW: Perform operations on Swift packages

USAGE: swift package [command] [options]

COMMANDS:

init [--type <type>] (type: empty library executable system-module)	Initialize a new package
fetch	Fetch package dependencies
update	Update package dependencies
generate-xcodeproj [--output <path>]	Generates an Xcode project
show-dependencies [--format <format>] (format: text dot json)	Print the resolved dependency graph
dump-package [--input <path>]	Print parsed Package.swift as JSON

OPTIONS:

-C, --chdir <path>	Change working directory before any other operation
--color <mode>	Specify color mode (auto always never)
--enable-code-coverage	Enable code coverage in generated Xcode projects
-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

```
1 module CMongoC [system] {  
2     header "mongoc.h"  
3     link "mongoc-1.0"  
4     export *  
5 }
```

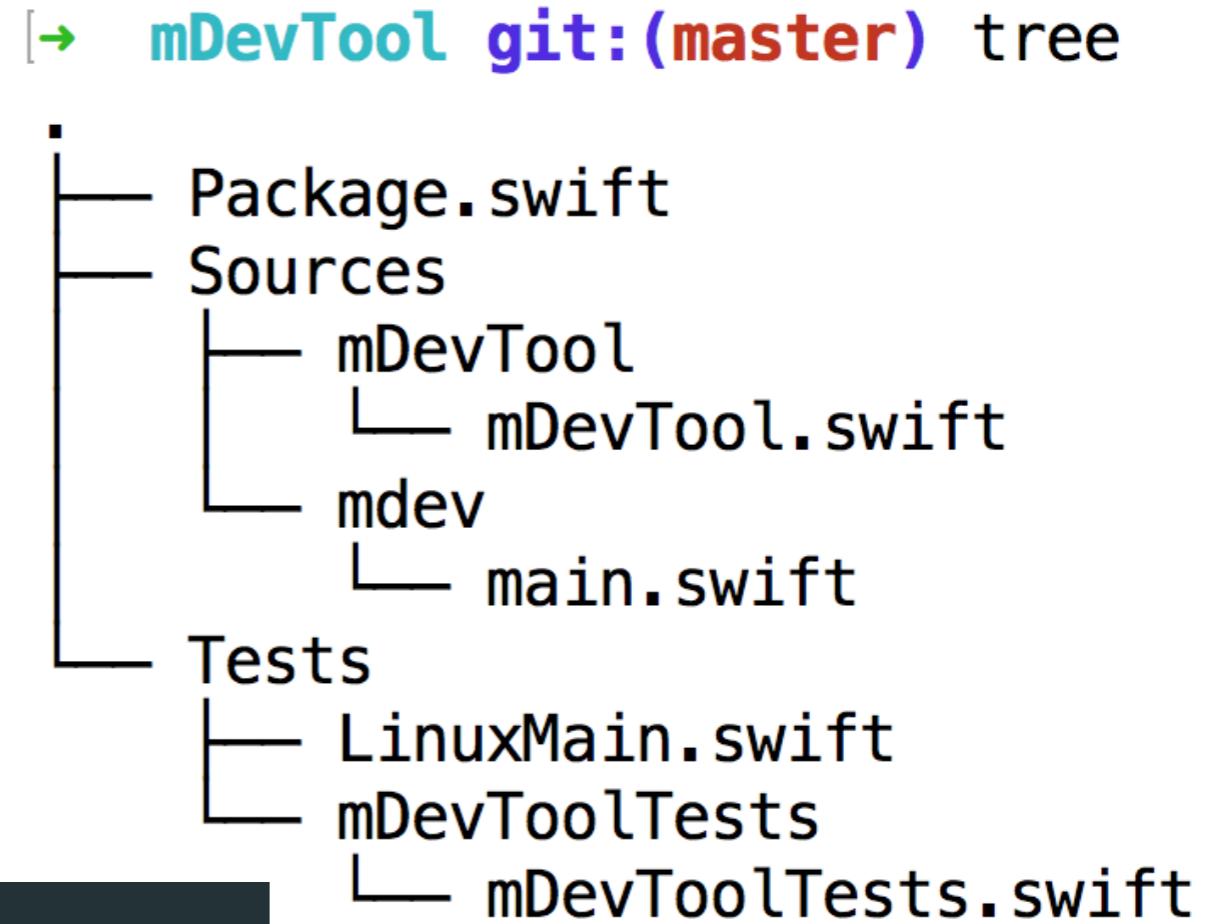
- how to use:

- <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

Explicit

Implicit



```
< > Package.swift ×

1 import PackageDescription

2

3 let package = Package(
4     name: "mDevTool",
5     targets: [
6         Target(name: "mdev", dependencies: ["mDevTool"])
7     ]
8 )
9
```

Dependencies



A screenshot of a code editor showing the `Package.swift` file. The file defines a package named "mDevServer" with two dependencies: "vapor/vapor" and "mDevTool". The code is numbered from 1 to 10.

```
1 import PackageDescription
2
3 let package = Package(
4     name: "mDevServer",
5     dependencies: [
6         .Package(url: "https://github.com/vapor/vapor.git", majorVersion: 1),
7         .Package(url: ".../mDevTool", majorVersion: 2)
8     ]
9 )
10
```

swift package show-dependencies

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

```
└ Vapor<https://github.com/vapor/vapor.git@1.0.2>
  └ Crypto<https://github.com/vapor/crypto.git@1.0.0>
    └ Core<https://github.com/vapor/core.git@1.0.0>
    └ CLibreSSL<https://github.com/vapor/clibressl.git@1.0.0>
  └ Fluent<https://github.com/vapor/fluent.git@1.0.0>
    └ Node<https://github.com/vapor/node.git@1.0.0>
      └ PathIndexable<https://github.com/vapor/path-indexable.git@1.0.0>
      └ Polymorphic<https://github.com/vapor/polymorphic.git@1.0.0>
  └ Engine<https://github.com/vapor/engine.git@1.0.0>
    └ Crypto<https://github.com/vapor/crypto.git@1.0.0>
      └ Core<https://github.com/vapor/core.git@1.0.0>
      └ CLibreSSL<https://github.com/vapor/clibressl.git@1.0.0>
    └ TLS<https://github.com/vapor/tls.git@1.0.0>
      └ CLibreSSL<https://github.com/vapor/clibressl.git@1.0.0>
      └ Core<https://github.com/vapor/core.git@1.0.0>
      └ Socks<https://github.com/vapor/socks.git@1.0.1>
    └ Core<https://github.com/vapor/core.git@1.0.0>
  └ Console<https://github.com/vapor/console.git@1.0.0>
    └ Polymorphic<https://github.com/vapor/polymorphic.git@1.0.0>
    └ Core<https://github.com/vapor/core.git@1.0.0>
  └ JSON<https://github.com/vapor/json.git@1.0.0>
```

<http://swiftpm-deps.honza.tech>

SwiftPM Dependency Viewer

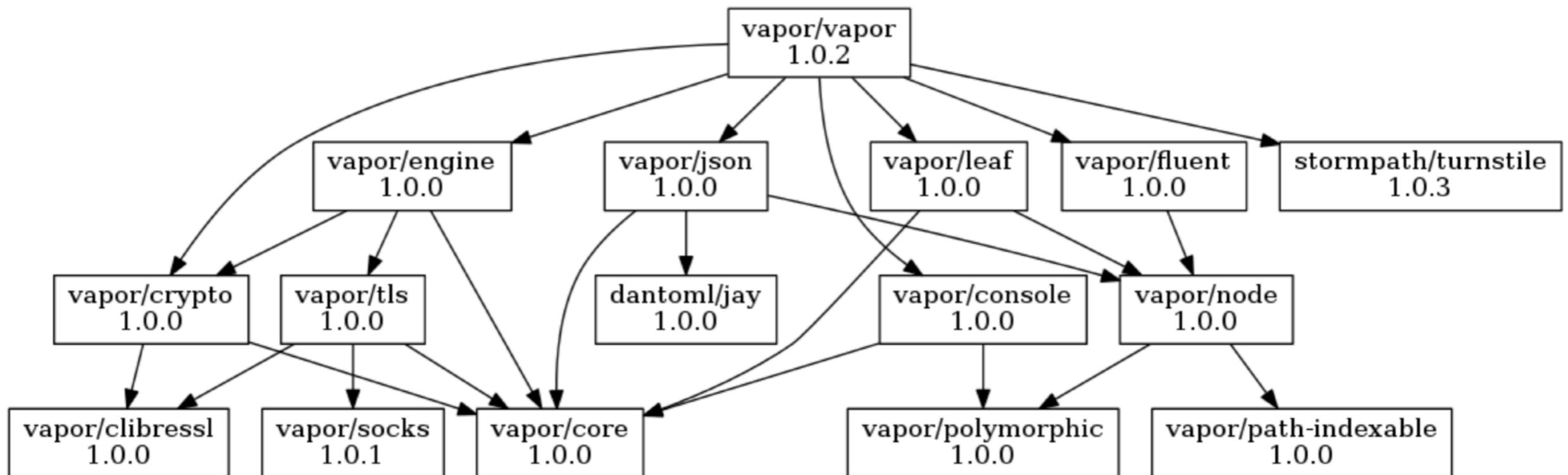
[Source & Documentation](#)

SwiftPM Repo GitHub name:

Tag (optional, latest by default):

PNG Graph Interactive Graph JSON

[View dependencies](#) 





SwiftPM Evolution

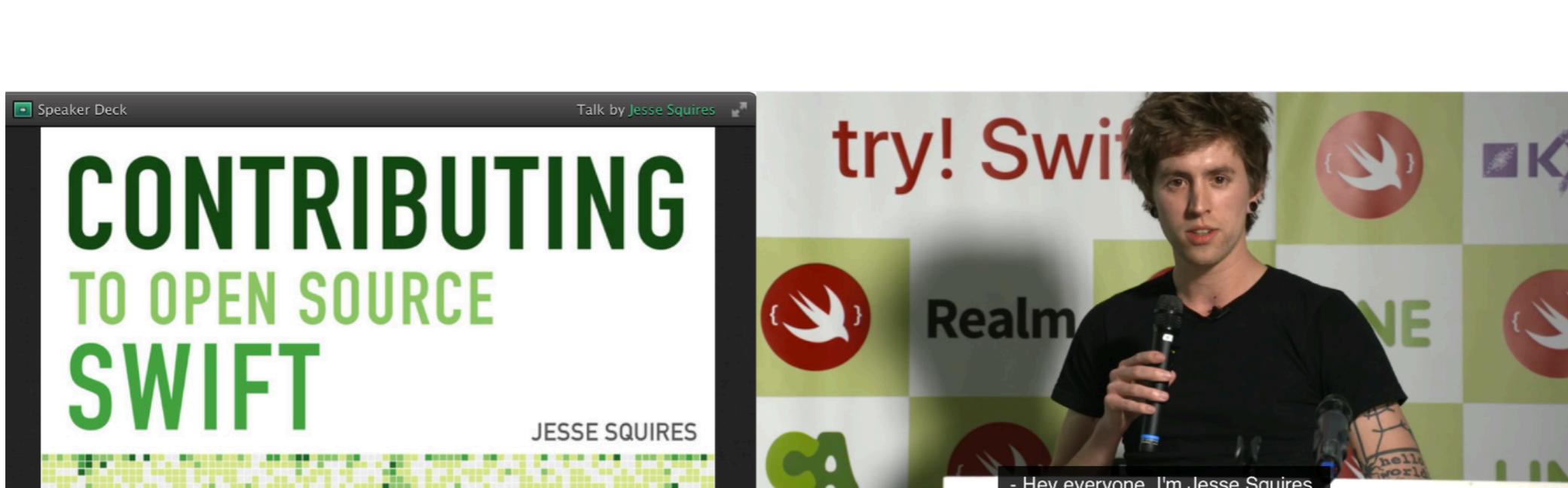
- Community Proposal
- SE-0019 Package Manager Testing
- SE-0038 C Language Target Support
- SE-0063 System Module Search Paths
- SE-0082 Editable Packages
- SE-0085 Command Names
- SE-0129 Test Naming Conventions
- SE-0135 Support for Differentiating Packages by Swift version

Missing features

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

Contributing to OSS Swift

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



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/2016-09-29-00-mdevtalk_swift_package_manager



WWW.MDEVTALK.CZ

mdevtalk