



&



Megan Kostick
18 February 2017

AGENDA

- Introduction
- SPM
- Kitura
- Demo
 - Creating a RESTful backend in Swift
- Demo
 - AngularJS App using Kitura
- Questions
- Resources

MEGAN KOSTICK



Central Washington University
B.S. Computer Science



Software Developer
IBM 4+ years



Swift @ IBM / Develop Advocacy
Apple Partnership & Swift Package Catalog

SPM

- Swift Package Manager
- <https://github.com/apple/swift-package-manager>
- Package.swift:

```
import PackageDescription

let package = Package(
    name: "CatalogSearch",
    dependencies: [
        .Package(url: "https://github.com/IBM-Swift/Kitura.git", majorVersion: 1, minor: 1),
        .Package(url: "https://github.com/IBM-Swift/Kitura-Cache.git", majorVersion: 1, minor: 1),
        .Package(url: "https://github.com/IBM-Swift/HeliumLogger.git", majorVersion: 1, minor: 1),
    ]
)
```

KITURA

- www.kitura.io
- A Swift web framework and HTTP server
- Open Sourced by IBM
- Currently compatible with Swift 3.0.2 GA



DEMO - KITURA

- Download Xcode 8.2.1 from the App Store



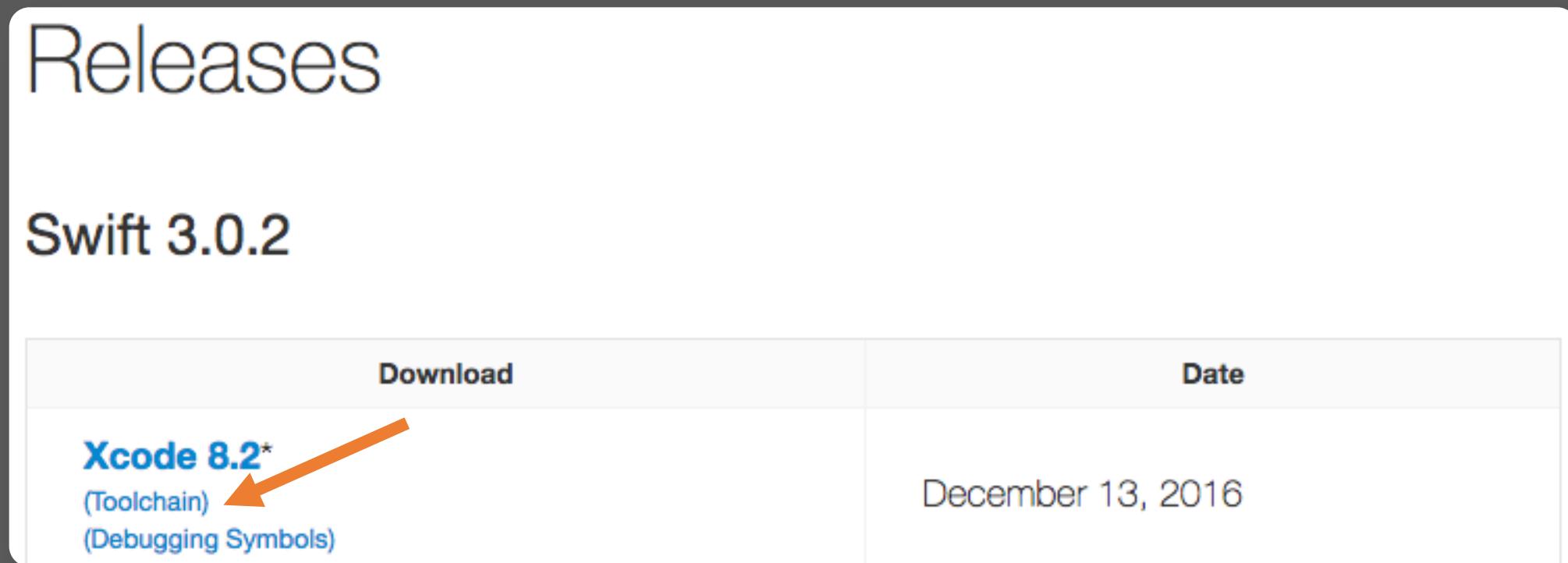
DEMO - KITURA

- Download and install the Swift 3.0.2 Toolchain from swift.org/download/:

Releases

Swift 3.0.2

Download	Date
Xcode 8.2* (Toolchain) (Debugging Symbols)	December 13, 2016



DEMO - KITURA

- Add Swift Toolchain to your PATH:

```
export PATH=/Library/Developer/Toolchains/swift-3.0.2-RELEASE.xctoolchain/usr/bin/:$PATH
```

DEMO - KITURA

- Create a new directory for your project:

```
mkdir myFirstProject
```

- Move into the newly created directory:

```
cd myFirstProject
```

- Initialize a new Swift project using SPM:

```
swift package init --type executable
```

DEMO - KITURA

- <https://github.com/kostickm/KituraHelloWorld/demoPart1/>
- In Package.swift, add Kitura as a dependency for your project:

```
import PackageDescription

let package = Package(
    name: "myFirstProject",
    dependencies: [
        .Package(url: "https://github.com/IBM-Swift/Kitura.git", majorVersion: 1, minor: 6)
    ]
)
```

DEMO - KITURA

- In Sources/main.swift, add the following code:

```
import Kitura

// Create a new router
let router = Router()

// Handle HTTP GET requests to /
router.get("/") {
    request, response, next in
    response.send("Hello, World!")
    next()
}

// Add an HTTP server and connect it to the router
Kitura.addHTTPServer(onPort: 8090, with: router)

// Start the Kitura runloop (this call never returns)
Kitura.run()
```

DEMO - KITURA

- **Compile your application:**

swift build

- **Run your new web application:**

.build/debug/myFirstProject

- **Open your browser and visit <http://localhost:8090>**

DEMO - KITURA

- <https://github.com/kostickm/KituraHelloWorld/demoPart2/>
- In Package.swift, add HeliumLogger as a dependency for your project:

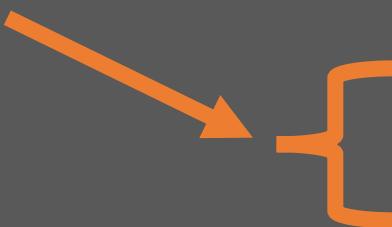
```
import PackageDescription

let package = Package(
    name: "myFirstProject",
    dependencies: [
        .Package(url: "https://github.com/IBM-Swift/Kitura.git", majorVersion: 1, minor: 6),
        .Package(url: "https://github.com/IBM-Swift/HeiliumLogger.git", majorVersion: 1, minor: 6)
    ]
)
```



DEMO - KITURA

- In Sources/main.swift, enable HeliumLogger:



```
import Kitura
import HeliumLogger

// Initialize HeliumLogger
HeliumLogger.use()

// Create a new router
let router = Router()

// Handle HTTP GET requests to /
router.get("/") {
    request, response, next in
    response.send("Hello, World!")
    next()
}

// Add an HTTP server and connect it to the router
Kitura.addHTTPServer(onPort: 8090, with: router)

// Start the Kitura runloop (this call never returns)
Kitura.run()
```

DEMO - KITURA

- Recompile your application:

swift build

- Run your new web application:

.build/debug/myFirstProject

- Open your browser and visit <http://localhost:8090>

- Now you will see logging in your terminal

DEMO - KITURA

- <https://github.com/kostickm/KituraHelloWorld/demoPart3/>
- In Sources/main.swift, add a new route:

```
// Status route
router.get("/status") {
    request, response, next in
    response.send("OK")
    next()
}
```

DEMO - KITURA

- Recompile your application:

swift build

- Run your new web application:

.build/debug/myFirstProject

- Open your browser and visit <http://localhost:8090/status>

DEMO - KITURA

- Turn your project into an Xcode project:

```
/Library/Developer/Toolchains/swift-3.0.2-RELEASE.xctoolchain/usr/bin/swift  
package generate-xcodeproj -Xswiftc -I./.build/debug -Xlinker -  
L./.build/debug
```

DEMO – AngularJS / Kitura

- <https://github.com/IBM-Swift/Kitura-AngularJS-Sample>

Questions ???



RESOURCES

- Swift: swift.org
- Kitura: kitura.io
- GitHub: [kostickm](https://github.com/kostickm)
- LinkedIn: www.linkedin.com/in/kostickmegan
- Twitter: [@kostickmegan](https://twitter.com/kostickmegan)