

Explore ... can we leverage our key learnings?

Server Side Swift

Andrew Halls

We must develop knowledge optimization initiatives to leverage our key learnings.

—Dilbert

Ski Instructor | App Developer | Mentor at thinkful.com

Software development is where people meet, to bend technology.



Andrew Halls



Meetup since 2011 | 6 Years Time to Give Back |
Not so much a geek, but a student of behavior and how we learn
| Helping People Find Joy
| Learning starts with Fun

Outline

Explore Why: Swift on the Server

Walk-through building and deploying a Server Written in Swift

Other Random Thoughts



Key technologies IBM's Kitura Web Framework written in Swift
Docker ... it was the delightful discover that made hosting work well

What Problem Are We Solving on the Server

Persistence / Sharing

Communications

Big Data | Machine
Learning | Cognitive
Services

Because You're there
(CLI)



App Admin

Remember why?

Database Server & Website

Apache

HTML

MySQL

PHP



It was weird, but server Side use to be easy.



Server's have gotten amazingly complex



We are just programmers and app developers.
"I just want to code"

Can't we do meaningful stuff and just do what we do best and code?

Can it Be Simple Again?



Swift
Xcode
Docker

Why Swift

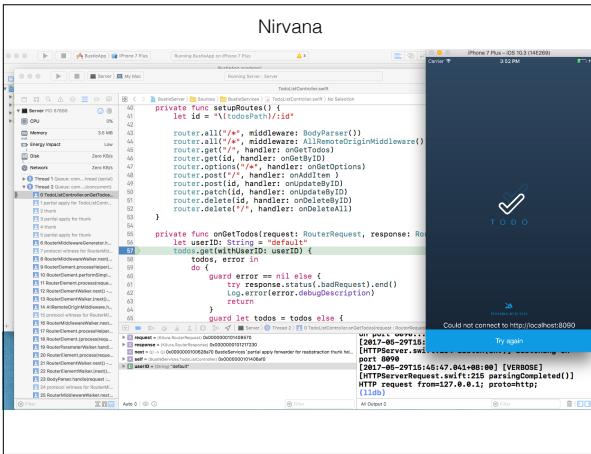
Frontend|Backend Same: Language, Libraries and Tools

Compiled

Performance

Strongly Typed

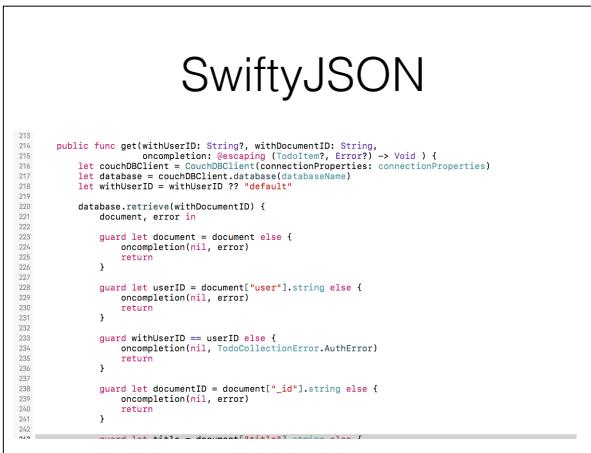
It's, well, Swifty



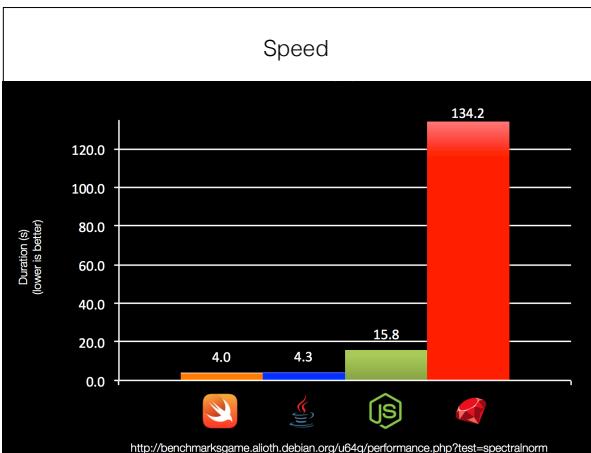
Server Xcode

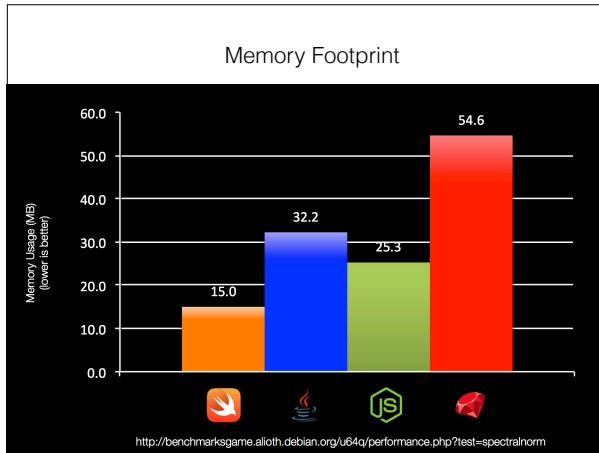
App Xcode Emulator

Full Debugger, Shown break point in Xcode, debugging my server, Truly Nirvana!



Common frameworks



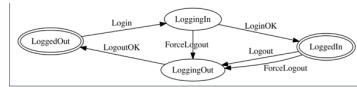


Other languages are Compiled and out perform Rub and Javascript significantly



Audience participation ?

Type Rich DSL



```
// 2. Setup state-transition mappings.  
let mappings: [Automaton<State, Input>.EffectMapping] = [  
  
    /* Input | fromState => toState | Effect */  
    /* ----- */  
    .login | .loggedOut => .loggingIn | loginOKProducer,  
    .loginOK | .loggingIn => .loggedIn | empty(),  
    .logout | .loggedIn => .loggedOut | logoutOKProducer,  
    .logoutOK | .loggedOut => .loggedOut | empty(),  
  
    .forceLogout | canForceLogout => .loggingOut | forceLogoutOKProducer  
]
```

Example of why I think Swift is cool ... you can build DSL's

What is the Dev Experience Right Now!

IBM's Kitura

Build & Test in Xcode

Test on Linux, Locally with Docker

Deploy with Docker

It was surprisingly pleasant ... Not only Swift but Docker made it possible



Lots happening in the WebServices in Swift world.

Google your favorite Javascript framework with Swift you'll find a project

I chose Kitura, Good tutorials and a book

HTTP Server (Don't Need Apache)

Web Framework Modeled after Express

URL routing, Middleware, Static File Service, etc ...

Project Organization

```
1 import PackageDescription
2
3 let package = Package(
4     name: "BustleServer",
5     targets: [
6         Target(
7             name: "Server",
8             dependencies: [.Target(name: "BustleServices")]
9         ),
10        Target(
11            name: "BustleServices"
12        )
13    ],
14    dependencies: [
15        .Package(url: "https://github.com/IBM-Swift/Kitura.git",
16                 .Package(url: "https://github.com/davidungar/minPromiseKit",
17                 .Package(url: "https://github.com/IBM-Swift/Kitura-CouchDB.git",
18                         .Package(url: "https://github.com/IBM-Swift/CloudConfiguration",
19                                 .Package(url: "https://github.com/IBM-Bluemix/cf-deployment-t
20                                     majorVersion: 3)
21     ]
22 )
```

SPM Project Server Main Unit Tests

IBM Package Catalog

Search for swift packages, submitter name, topics, git url, etc.

HeliumLogger
By IBM-Swift

README.md

HeliumLogger

Provides a lightweight Swift Logging framework.

Features:

- Different logging levels such as Warning, Verbose, and Error
- Color output to terminal

Usage:

To include this in your code paste this into your package.swift file

```
Package(url: "https://github.com/IBM-Swift/HeliumLogger.git")
```

Copied!

Find and review your components
Apple SPM Code To copy
Test in the Sandbox

IBM Swift Sandbox

Swift Ver. 3.8 (HeliumLogger)

```
1 // Copyright IBM Corporation 2015
2 // Licensed under the Apache License, Version 2.0 (the "License");
3 // you may not use this file except in compliance with the License.
4 // You may obtain a copy of the License at
5 // http://www.apache.org/licenses/LICENSE-2.0
6 //
7 // Unless required by applicable law or agreed to in writing, software
8 // distributed under the License is distributed on an "AS IS" BASIS,
9 // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
10 // See the License for the specific language governing permissions and
11 // limitations under the License.
12
13 import HeliumLogger
14 import Logging
15
16 Log.debug("This is a verbose log message")
17 Log.info("This is an informational log message")
18 Log.warning("This is a warning")
19 Log.error("This is an error")
20 Log.debug("This is a debug message")
21 let type = LoggerMessageType.info
22 Log.debug("This is a dynamic message", functionName: "no func", lineNum: #line,
23 fileNum: #file)
```

New Load Code Save Code ▶ Settings

IBM Swift Sandbox

Try out the code in an web based environment

```
swift build
```

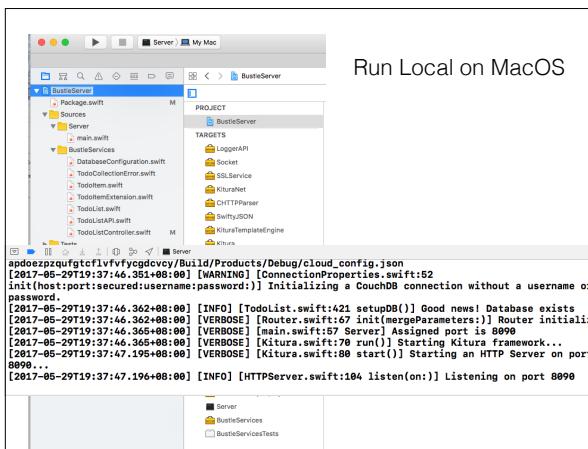
```
GALTMB03:BustleServer ahalls$ swift build
Compile Swift Module 'Socket' (3 sources)
Compile Swift Module 'LoggerAPI' (1 sources)
Compile Swift Module 'SwiftyJSON' (2 sources)
Compile Swift Module 'MiniPromiseKit' (4 sources)
Compile Swift Module 'KituraTemplateEngine' (1 sources)
Compile Swift Module 'HeliumLogger' (2 sources)
Compile Swift Module 'Configuration' (6 sources)
Compile Swift Module 'TestProgram' (1 sources)
Compile Swift Module 'CloudFoundryEnv' (6 sources)
Linking ./build/debug/TestProgram
Compile Swift Module 'TokuConfig' (1 sources)
Compile Swift Module 'CloudFoundryConfig' (2 sources)
Compile Swift Module 'AmazonConfig' (1 sources)
Compile Swift Module 'SSLService' (1 sources)
Compile CHTTPParser utils.c
Compile CHTTPParser http_parser.c
Linking ./build/debug/CHTTPParser
Compile Swift Module 'KituraNet' (34 sources)
Compile Swift Module 'CouchDB' (5 sources)
Compile Swift Module 'Kitura' (43 sources)
Compile Swift Module 'CloudFoundryDeploymentTracker' (1 sources)
Compile Swift Module 'CouchDBSample' (1 sources)
Linking ./build/debug/CouchDBSample
Compile Swift Module 'BustleServices' (7 sources)
Compile Swift Module 'Server' (1 sources)
Linking ./build/debug/Server
GALTMB03:BustleServer ahalls$
```

build the app

```
swift generate-xcodeproj
```

```
GALTMB03:BustleServer ahalls$ swift package generate-xcodeproj
generated: ./BustleServer.xcodeproj
GALTMB03:BustleServer ahalls$
```

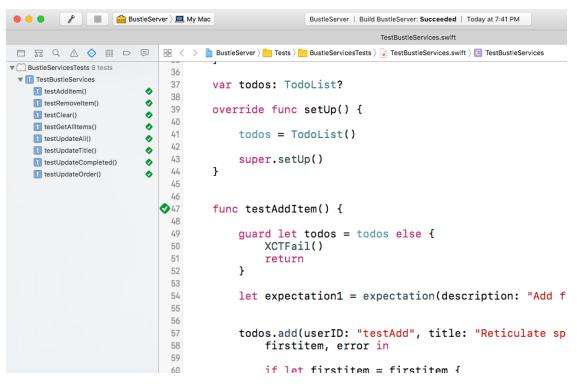
Generate Xcode Project



Run Local on MacOS

Package manager generates a lot of targets, and you usually pointed to the wrong one

Run Unit Test Local on MacOS

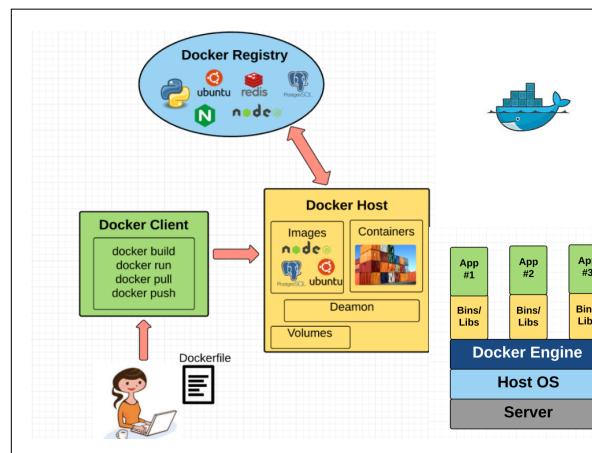


A screenshot of an Xcode interface showing a Swift file named `TestBustleServices.swift`. The code is for a unit test module named `BustleServicesTests`. It contains several test cases for methods like `testAddItem`, `testUpdateItem`, and `testUpdateCompleted`. The code uses XCTest framework for assertions.

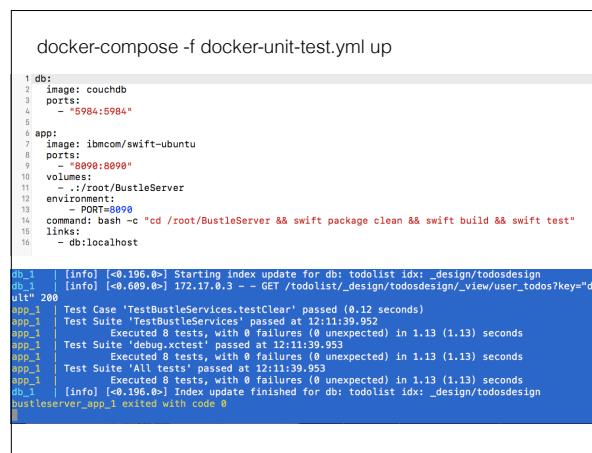
```
var todos: TodoList?
override func setUp() {
    todos = TodoList()
    super.setUp()
}

func testAddItem() {
    guard let todos = todos else {
        XCTFail()
        return
    }
    let expectation1 = expectation(description: "Add first item")
    todos.add(userID: "testAdd", title: "Reticulate splines", error: nil)
    if let firstItem = firstItem {
        XCTAssertEqual(firstItem.title, "Reticulate splines")
        expectation1.fulfill()
    }
    wait(for: [expectation1], timeout: 1)
}
```

Can't run Swift Unit Test on Main Module ... that is why most of the server code is in a separate module.



Docker is DevOps for mere mortals



A terminal window showing the command `docker-compose -f docker-unit-test.yml up` being run. The output shows the creation of a database container (`db`) and an application container (`app`). The application container runs a script to clean, build, and test the application. The log ends with the message `bustleserver_app_1 exited with code 0`.

```
1 db:
2   image: couchdb
3   ports:
4     - "5984:5984"
5 
6 app:
7   image: ibmcom/swift-ubuntu
8   ports:
9     - "8990:8990"
10  volumes:
11    - ./root/BustleServer
12  environment:
13    PORT=8990
14  command: bash -c "cd /root/BustleServer && swift package clean && swift build && swift test"
15  links:
16    - db:localhost

db_1 | [info] <0.196.0> Starting index update for db: todolist idx: _design/todosdesign
db_1 | [info] <0.609.0> -- GET /todolist/_design/todosdesign/_view/user_todos?key="defa
ult" 200
db_1 | [info] <0.196.0> Test Case 'TestBustleServices.recycle' passed (0.12 seconds)
app_1 | [info] <0.196.0> Test Suite 'TestBustleServices' passed at 12:11:39.952
app_1 | [info] <0.196.0> Executed 8 tests, with 0 failures (0 unexpected) in 1.13 (1.13) seconds
app_1 | [info] <0.196.0> Test Suite 'debug.xctest' passed at 12:11:39.953
app_1 | [info] <0.196.0> Executed 8 tests, with 0 failures (0 unexpected) in 1.13 (1.13) seconds
app_1 | [info] <0.196.0> Test Suite 'All tests' passed at 12:11:39.953
app_1 | [info] <0.196.0> Executed 8 tests, with 0 failures (0 unexpected) in 1.13 (1.13) seconds
[info] <0.196.0> Index update finished for db: todolist idx: _design/todosdesign
bustleserver_app_1 exited with code 0
```

Unit Tests
From your App or Postman

Deploy on Bluemix

```
[GALTM803:BustleServer ahalls$ cf push
Using manifest file /Users/ahalls/Work/Galst.yml
Updating app bustle-server in org galsoft
.
OK
Uploading bustle-server...
Uploading app files from: /Users/ahalls/Work/Galst.yml
Uploading 46.7K, 63 files
Done uploading
requested state: started
instances: 1/1
usage: 256M x 1 instances
urls: bustle-server.mybluemix.net
last uploaded: Tue May 30 08:41:14 UTC 2017
stack: cflinuxfs2
buildpack: swift_buildpack
```

Configure Server
Upload via cf push command

Where is Swift, Going Next

Command Line Tools

Serverless Servers, IBM OpenWhisk

Swift on Raspberry PI

Swift on Android

Swift on?

CLI | Swift 4.0, Strings as powerful as Perl
OpenWhisk ... build process still lumpy ... Makefiles?
Look for Swift in other environments

WWDC Wish List

Refactoring Tools for Swift

Full Featured Package Manager, for Xcode too

Foundation complete on Linux

String Manifesto meets it's goal, better than Perl

Build tools or hooks for "Serverless" delivery

Xcode Support for DSL's

slide dated ... definitely got the first one, have to look in the details for the others.

Resources

Swift@IBM
swift-at-ibm.slack.com



Server API Development in
Swift, Kitura, & Bluemix
Mark Price, Unity 3D Android iOS
10 Swift 3 & React Teacher

51% Complete  Your Rating



HACKING WITH SWIFT
SERVER-SIDE SWIFT
COMPLETE TUTORIAL COURSE
Learn to make web
apps with real-world
Swift projects
Paul Hudson

20% off any book

The Udemy course is amazing. In ~20 hours of video you watch him type every keystroke of a server in swift, app in swift and all the supporting configurations ... with a cogent running commentary, an amazing feet of live coding. He's a god ... don't let the southern accent throw you off!