# SWIFT服务器端编程：从入门到实践

shanks

2016-11-05

# 自我介绍

- 杨晖/shanksyang(github)
- SwiftGG翻译组成员，翻译组大中华区总管理员
- 曾就职于腾讯QQ会员，拍拍等部门，9年linux 后台开发经验
- 目前就职于深圳智美运动科技，从事体育行业的创业
- Swift 深度爱好者，崇尚开源精神，坚信 server-side swift 会逐步工业化

# AGENDA

- 给我一个 Swift 服务器端编程的理由
- Server-side Swift 现状
- Swift 后端编程必备知识
- 实践：使用vapor构建restful API demo
- 深入Swift 服务器端框架实现
- 后记

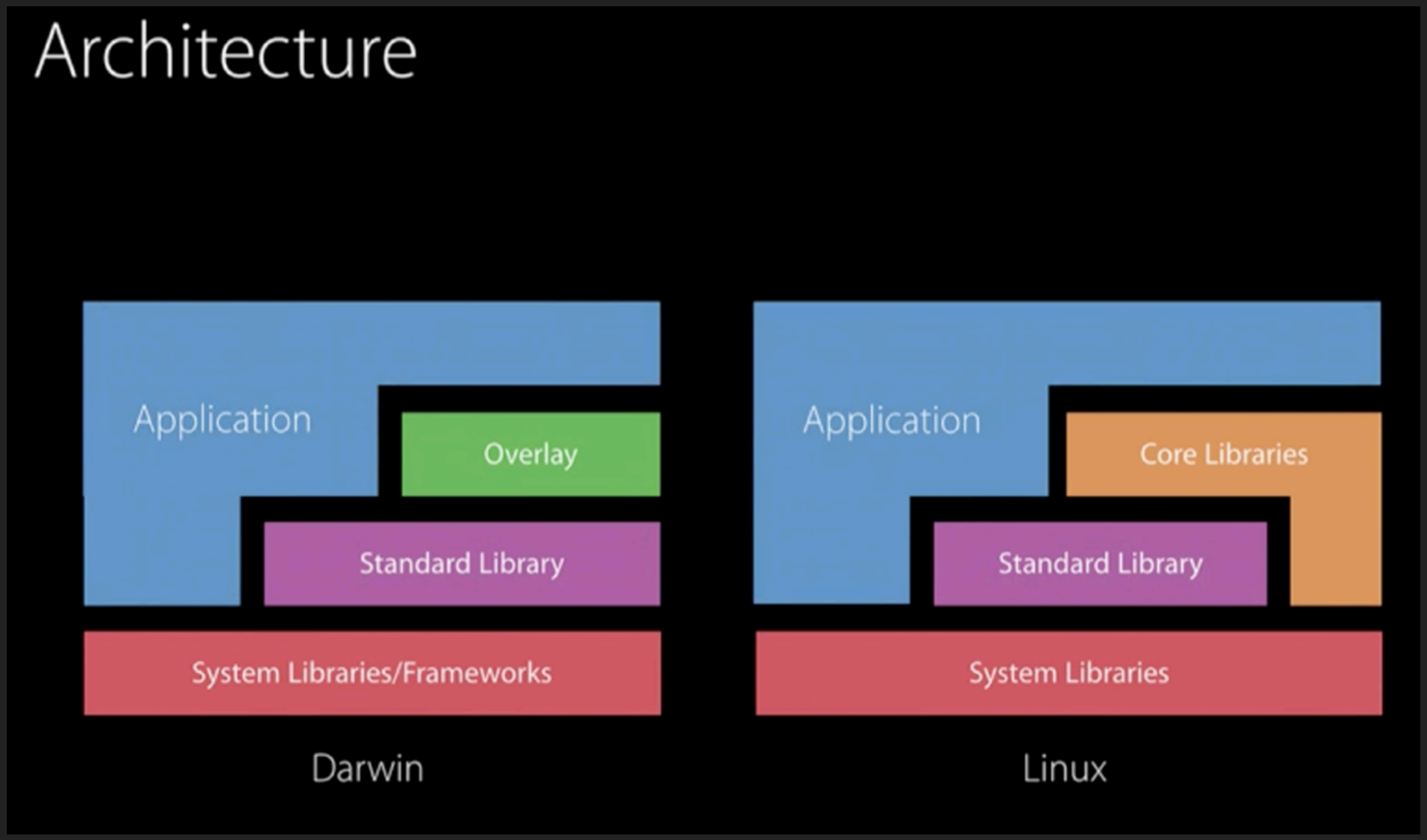
# 给我一个 **SWIFT** 服务器端编程的理由

- 开源！
- 开发效率高
- 性能卓越
- 社区活跃

# 开源

- Apache 2.0 license
  - 对商业应用友好的许可
- 开源的东西有那些？
  - The Swift compiler
  - The standard library
  - Core libraries
    - Foundation
    - GCD(libdispatch)
    - XCTest
  - Swift REPL and debugger (LLDB)
  - The Swift package manager

## 开源


Architecture
Application
Overlay
Standard Library
System Libraries/Frameworks
Darwin
Application
Core Libraries
Standard Library
System Libraries
Linux

## 官方支持平台

- MacOS
- Linux: Ubuntu 14.04, 15.10, 16.04

## 其他平台

- Swift for Windows
  - 目前还是停留在hello,world 阶段
- Swift for android
  - 目前只能使用swift标准库，不支持图形化界面
  - 据传 Google考虑使用Swift 作为Android的主要程序语言

## SWIFT 开源大事记

- 2015-12-03：正式开源
- swift-evolution:144 个 proposals
- 2016-09-13：正式发布 Swift 3.0 和 Swift PM
- 2016-10-25: Server APIs Work Group 成立

# 开发效率高

- 支持多种编程范式
  - Object-Oriented Programming
  - Functional Programming
  - Protocol Oriented programming
- 与客户端共享底层的开源 API 库：SwiftyJSON
- 安全的编码方式

## 性能卓越

- Swift 本身语言的效率
  - 编译语言
  - 尽量少动态绑定，动态派发(比较OC)
  - 强类型语言

## 社区活跃

- 大厂支持
  - Apple：Server APIs Project
  - IBM
    - Swift Sandbox（REPL online）
    - IBM Cloud Tools for Swift

## 社区活跃

- 框架和组件百花齐放
  - Web 框架：kitura, vapor, zewo, perfect
  - 通用库：SwiftyJSON, Stencil, Open Swift
- For backend：Redbird

# 服务器端WEB框架

KITURA™

A Swift Web Framework and HTTP Server

VAPOR

A Web Framework for Swift

Zewo

Build server software in Swift

Zewo is a set of open source libraries that help you build modern and blazing fast server software.

the perfect toolkit

MAKE YOUR SERVER-SIDE SWIFT CODE PERFECT

# 服务器端WEB框架

| 特性 | Kitura | Vapor | Perfect | Zewo |
|---|---|---|---|---|
| URL路由 | √ | √ | √ | √ |
| FastCGI | √ | × | √ | × |
| ORM | × | √ | √ | √ |
| 加密传输 | √ | √ | × | √ |
| 中间件 | √ | √ | √ | √ |
| WebSocket | √ | √ | √ | √ |
| 独立web服务器 | √ | √ | √ | √ |
| 模板渲染 | √ | √ | √ | √ |

# KITURA

- IBM 出品
- wwdc 2016 session 415
- IBM Cloud Tools for Swift

# VAPOR

- 灵感来自于PHP web 框架 Laravel/Lumen
- 文档齐全，开发入门首选
- 使用 ORM，性能一般

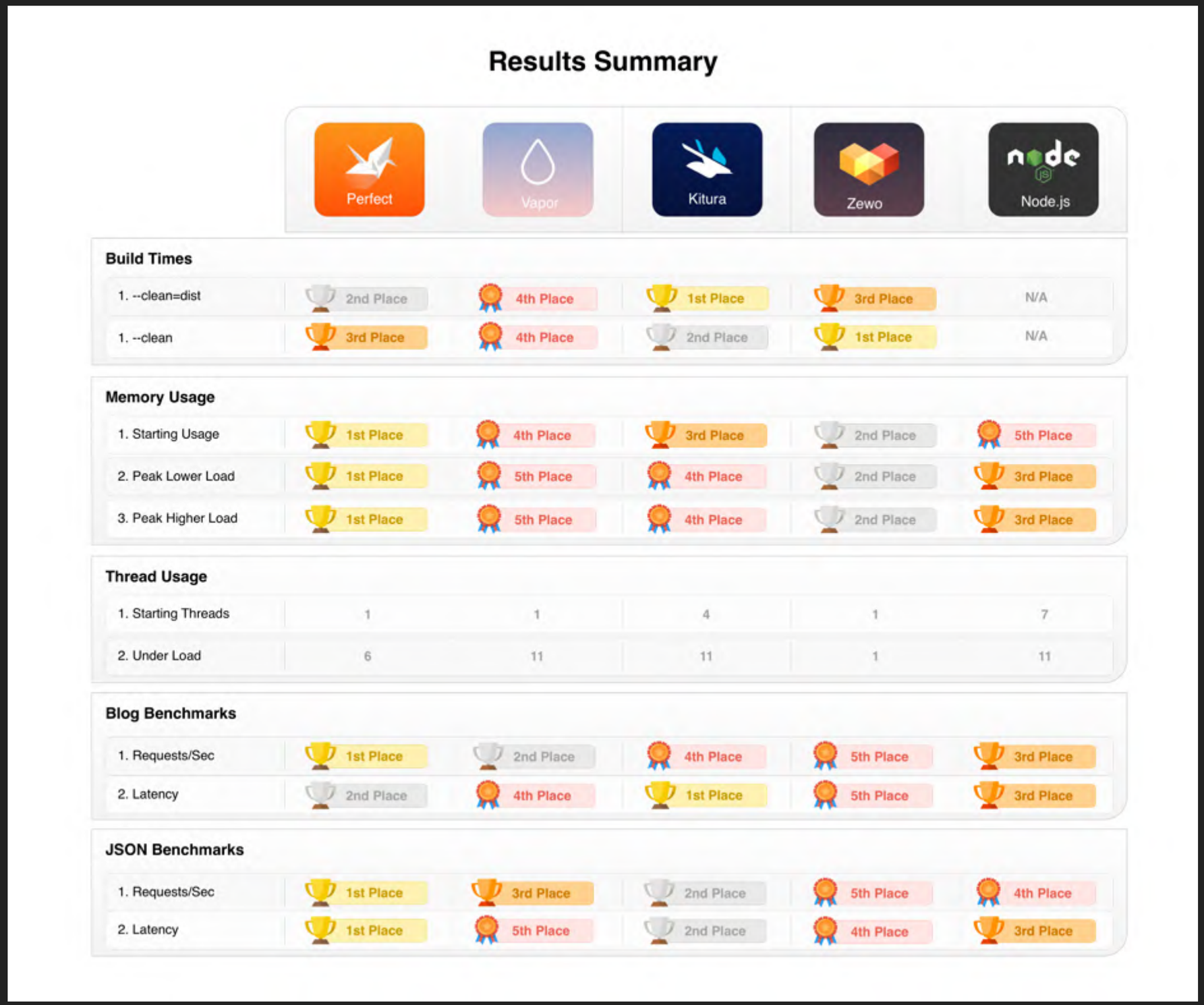
# PERFECT

- 目标是建立一套后台服务的生态系统，包罗万象
- 融资 150w，往商业化演进
- 代码风格类java和C#

# ZEWO

- Zewo的定位是一系列的Swift 库，用于构建Swift 后端应用
- 50多个 Swift Packages 组成，供其他web框架库使用
- 文档少

# SWIFT后端框架性能



## SWIFT 服务器端编程必备

- Swift PM
- swiftenv

# SWIFT PACKAGE MANAGER

- 方便在命令行下管理 Swift代码和跨平台编译
- 引入其他模块机制类似于Cocoapods和composer

# SWIFT PM使用

- 常用命令
  - 创建包：swift package init --type executable
  - 创建xcode 项目管理源代码:swift package generate-xcodeproj
- 编译包:swift build

## 构建步骤

- Ubuntu 下安装 Swift
- 新建项目
- 编写服务代码
- 编译
- 启动服务
- 客户端访问

# UBUNTU 下安装 SWIFT

- 安装 swiftenv

```
root@shanksubuntu1604-64:~# git clone https://github.com/kylef/swiftenv.git ~/.swiftenv
Cloning into '/root/.swiftenv'...
remote: Counting objects: 840, done.
remote: Total 840 (delta 0), reused 0 (delta 0), pack-reused 840
Receiving objects: 100% (840/840), 267.46 KiB | 0 bytes/s, done.
Resolving deltas: 100% (514/514), done.
Checking connectivity... done.
root@shanksubuntu1604-64:~# echo 'export SWIFTENV_ROOT="$HOME/.swiftenv"' >> ~/.bashrc
root@shanksubuntu1604-64:~# echo 'export PATH="$SWIFTENV_ROOT/bin:$PATH"' >> ~/.bashrc
root@shanksubuntu1604-64:~# echo 'eval "$(swiftenv init -)"' >> ~/.bashrc
```

## UBUNTU 下安装 SWIFT

- 安装 swift  3.0.1

```
root@shanksubuntu1604-64:~# swiftenv uninstall 3.0.1
root@shanksubuntu1604-64:~# swiftenv install 3.0.1
Downloading https://swift.org/builds/swift-3.0.1-release/ubuntu1604/swift-3.0.1-RELEASE/swift-3.0.1-RELEASE-ubuntu16
.04.tar.gz
3.0.1 has been installed.
root@shanksubuntu1604-64:~# swift -version
Swift version 3.0.1 (swift-3.0.1-RELEASE)
Target: x86_64-unknown-linux-gnu
root@shanksubuntu1604-64:~#
```

## 新建 SWIFT 服务器端项目

- swift package init --type executable

```
-> % swift package init --type executable
Creating executable package: idev_swiftsvr_sample
Creating Package.swift
Creating .gitignore
Creating Sources/
Creating Sources/main.swift
Creating Tests/
```

# 使用XCODE 来管理项目

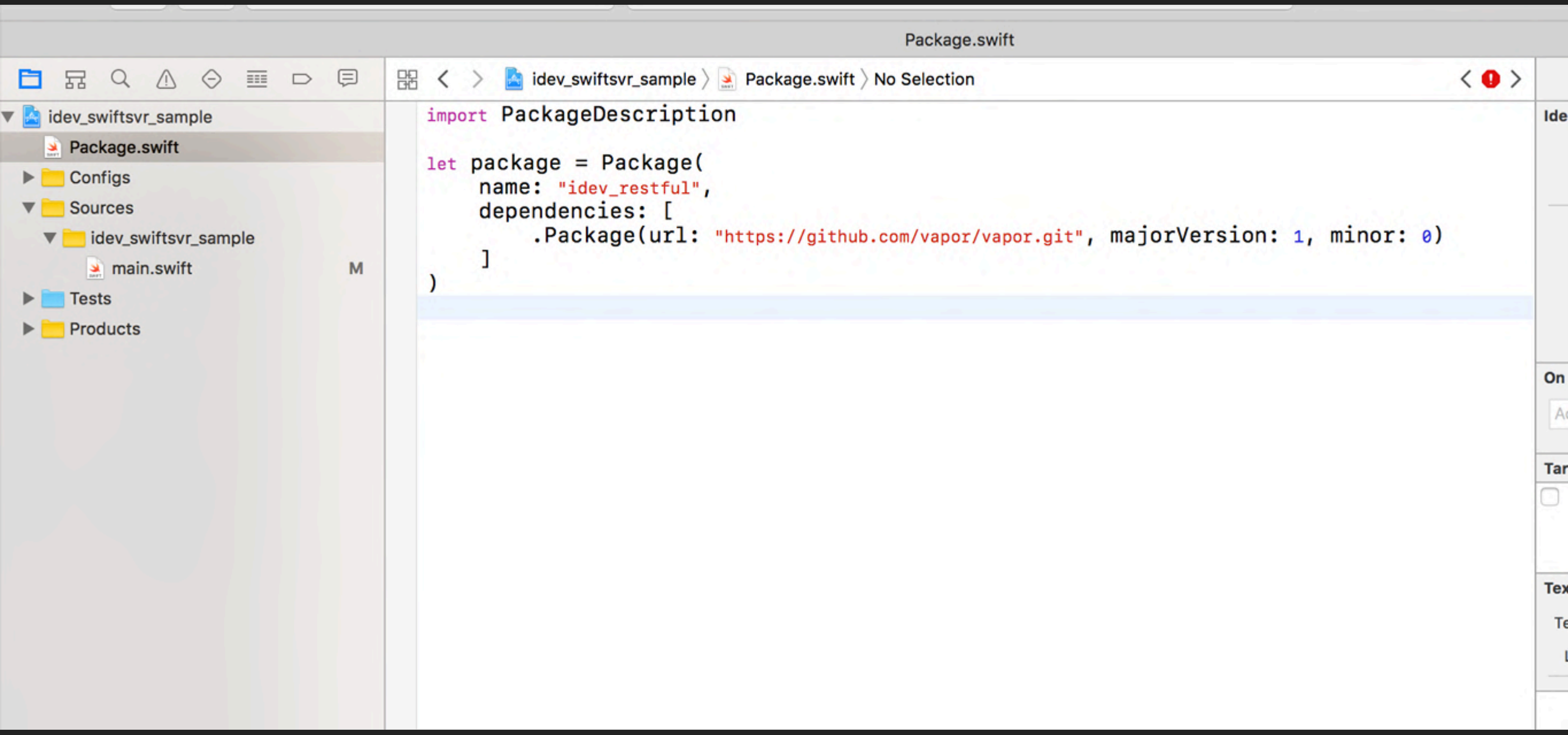
- swift package generate-xcodeproj

```
-> % swift package generate-xcodeproj
generated: ./idev_swiftsvr_sample.xcodeproj
```

# 实践：使用VAPOR 构建RESTFUL API DEMO

## 编写服务代码

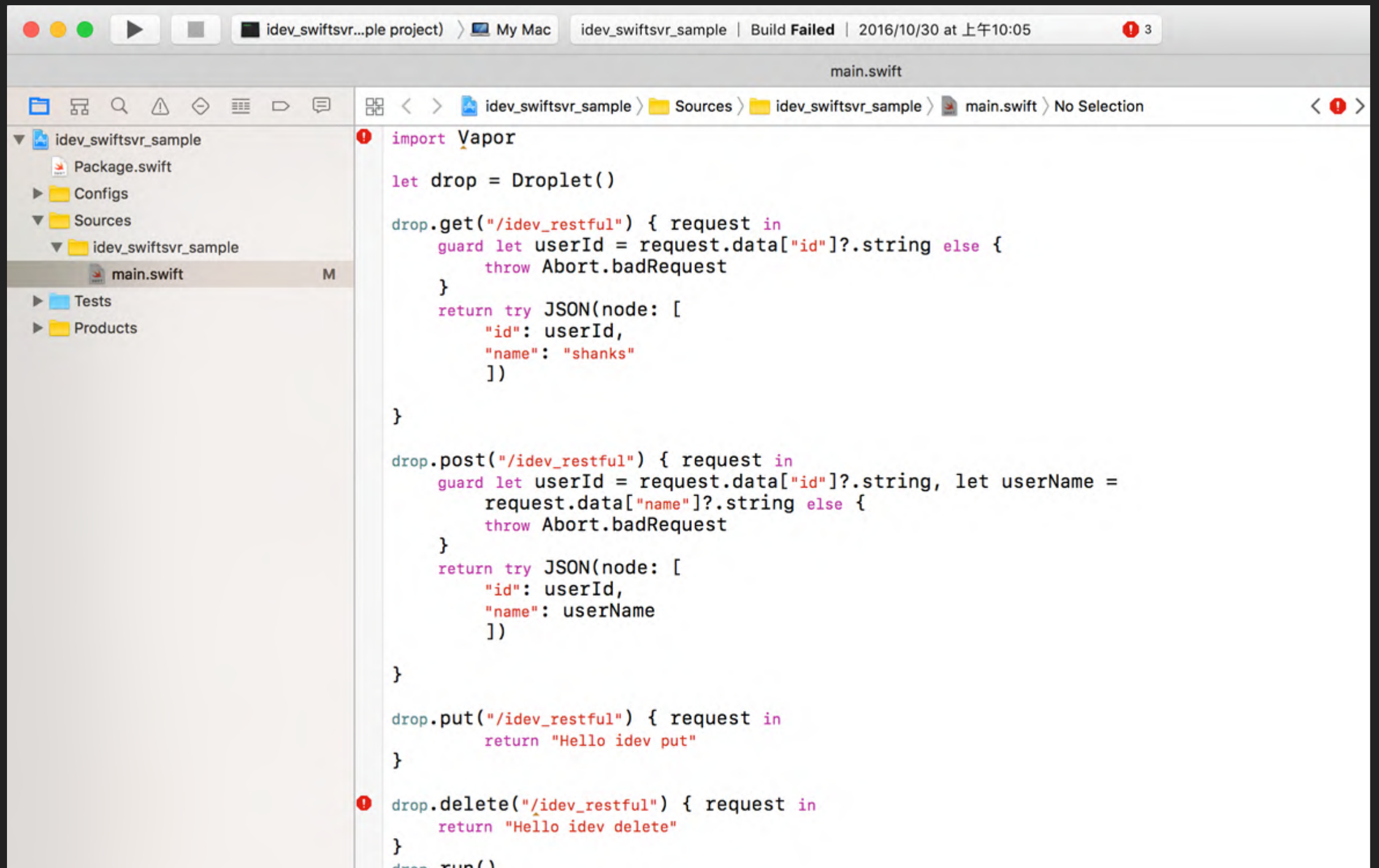
- 添加依赖



# 实践：使用VAPOR 构建RESTFUL API DEMO

## 编写服务代码

- 添加逻辑代码



## 编译代码

- swift build:第一次会下载依赖包，略慢

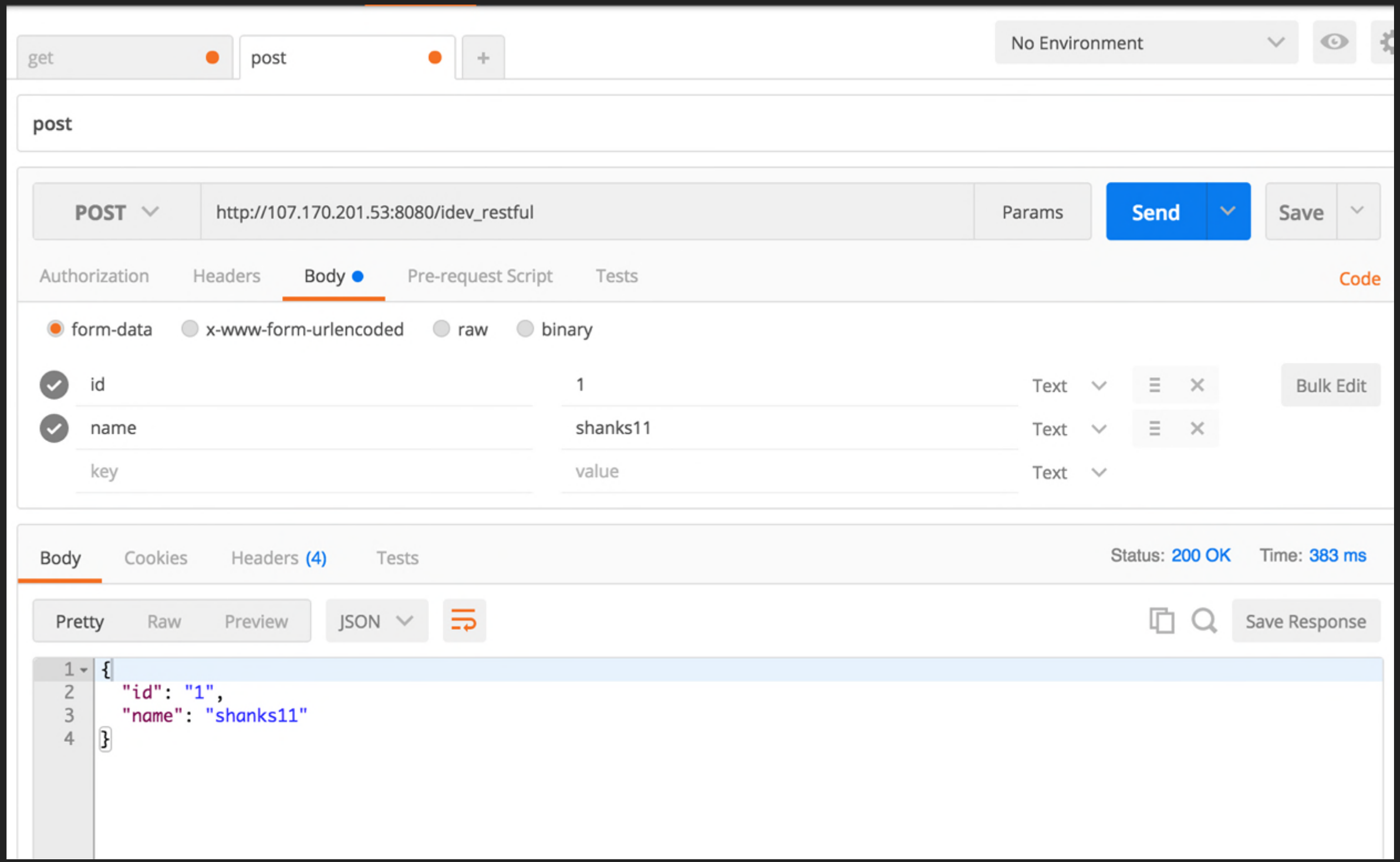
```
root@shanksubuntu1604-64:/data/code/idev_swiftsvr_sample# swift build
Compile Swift Module 'idev_restful' (1 sources)
Linking ./.build/debug/idev_restful
root@shanksubuntu1604-64:/data/code/idev_swiftsvr_sample#
```

## 运行服务

- .build/debug目录

```
root@shanksubuntu1604-64:/data/code/idev_swiftsvr_sample# .build/debug/idev_restful
Could not load localization files: Unreadable
No cipher key was set, using blank key.
Chacha20 cipher requires an initialization vector (iv).
No command supplied, defaulting to serve...
No preparations.
No servers.json configuration found.
Starting server at 0.0.0.0:8080
```

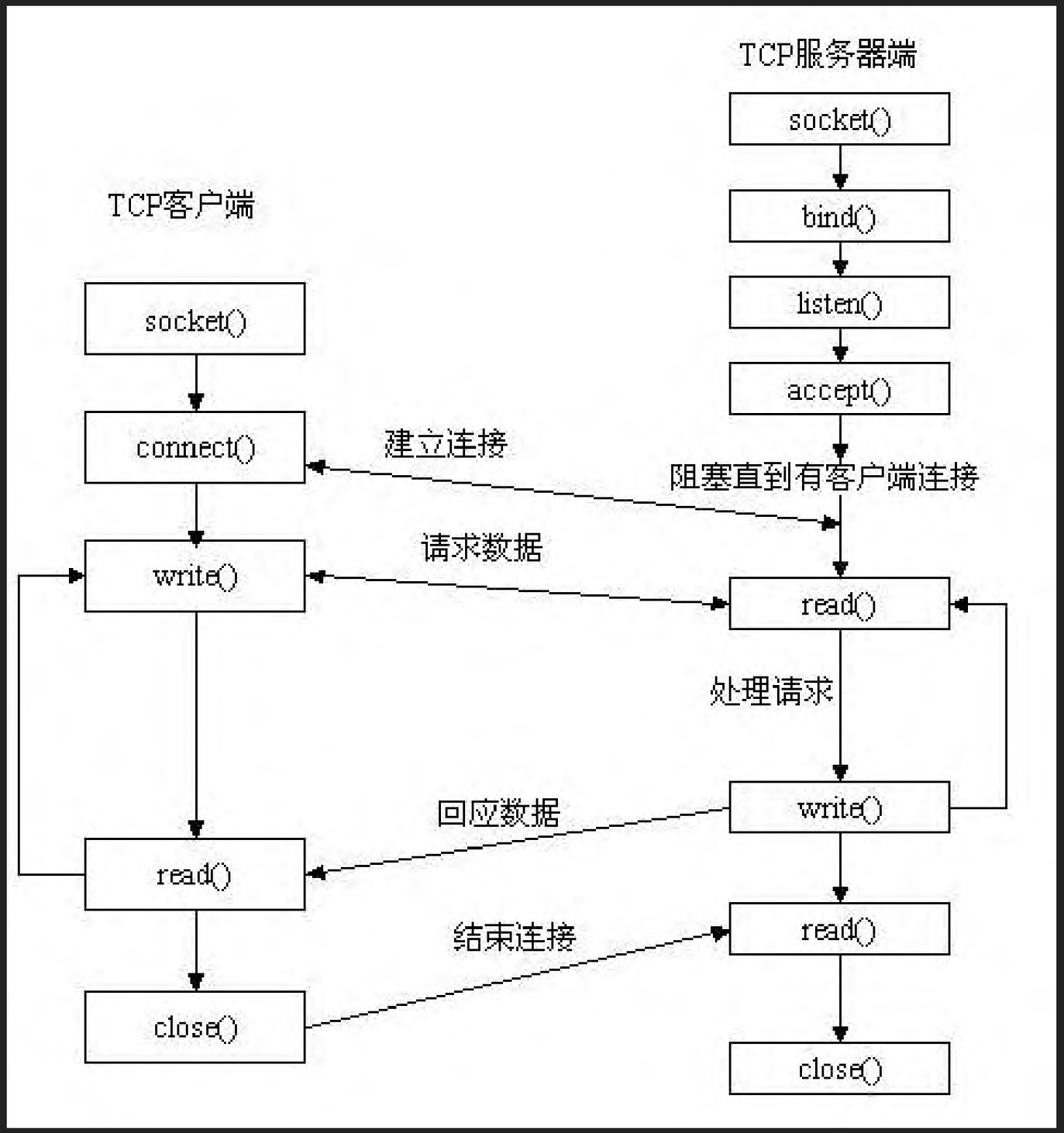
# 客户端访问


get
post
No Environment
post
POST
http://107.170.201.53:8080/idev_restful
Params
Send
Save
Authorization
Headers
Body
Pre-request Script
Tests
Code
form-data
x-www-form-urlencoded
raw
binary
id
1
Text
Bulk Edit
name
shanks11
Text
key
value
Text
Body
Cookies
Headers (4)
Tests
Status: 200 OK
Time: 383 ms
Pretty
Raw
Preview
JSON
Save Response
{
"id": "1",
"name": "shanks11"
}

# 深入框架实现

- Socket 编程
- 并发处理
- 进程间通信

# SOCKET 编程


TCP服务器端
socket()
bind()
listen()
accept()
阻塞直到有客户端连接
TCP客户端
socket()
connect()
建立连接
write()
请求数据
read()
处理请求
write()
回应数据
read()
read()
结束连接
close()
close()

## 并发处理

- 多进程
  - 进程间通信
- 多线程
  - Libdispatch(GCD) -> Kitura
- I/O 多路复用
  - Select -> Vapor
  - Poll
  - Epoll
  - Libevent -> Perfect
  - Libmill -> Zewo

## 进程间通信(IPC)

- 成熟的web框架下，需要有master进程和worker进程，就会涉及到某种形式下的进程间通信
- 常用分类
  - 管道
  - 消息队列
  - 共享内存
  - 套接字

## 后记

- 大规模应用的障碍
  - 完善底层API
  - 完善Swift PM
- 与nginx 和 apache 的结合

谢谢