

Chapter 50: Swift HTTP server by Kitura

Swift server with Kitura

Kitura is a web framework written in swift that is created for web services. It's very easy to set up for HTTP requests. For environment, it needs either OS X with XCode installed, or Linux running swift 3.0.

Section 50.1: Hello world application

Configuration

First, create a file called Package.swift. This is the file that tells swift compiler where the libraries are located. In this hello world example, we are using GitHub repos. We need Kitura and HeliumLogger. Put the following code inside Package.swift. It specified the name of the project as *kitura-helloworld* and also the dependency urls.

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

Next, create a folder called Sources. Inside, create a file called main.swift. This is the file that we implement all the logic for this application. Enter the following code into this main file.

Import libraries and enable logging

```
import Kitura
import Foundation
import HeliumLogger

HeliumLogger.use()
```

Adding a router. Router specifies a path, type, etc of the HTTP request. Here we are adding a GET request handler which prints *Hello world*, and then a post request that reads plain text from the request and then send it back.

```
let router = Router()

router.get("/get") {
    request, response, next in
    response.send("Hello, World!")
    next()
}

router.post("/post") {
    request, response, next in
    var string: String?
    do{
        string = try request.readString()

    } catch let error {
        string = error.localizedDescription
    }
    response.send("Value \(string!) received.")
    next()
}
```

```
}
```

Specify a port to run the service

```
let port = 8080
```

Bind the router and port together and add them as HTTP service

```
Kitura.addHTTPServer(onPort: port, with: router)  
Kitura.run()
```

Execute

Navigate to the root folder with Package.swift file and Resources folder. Run the following command. Swift compiler will automatically download the mentioned resources in Package.swift into Packages folder, and then compile these resources with main.swift

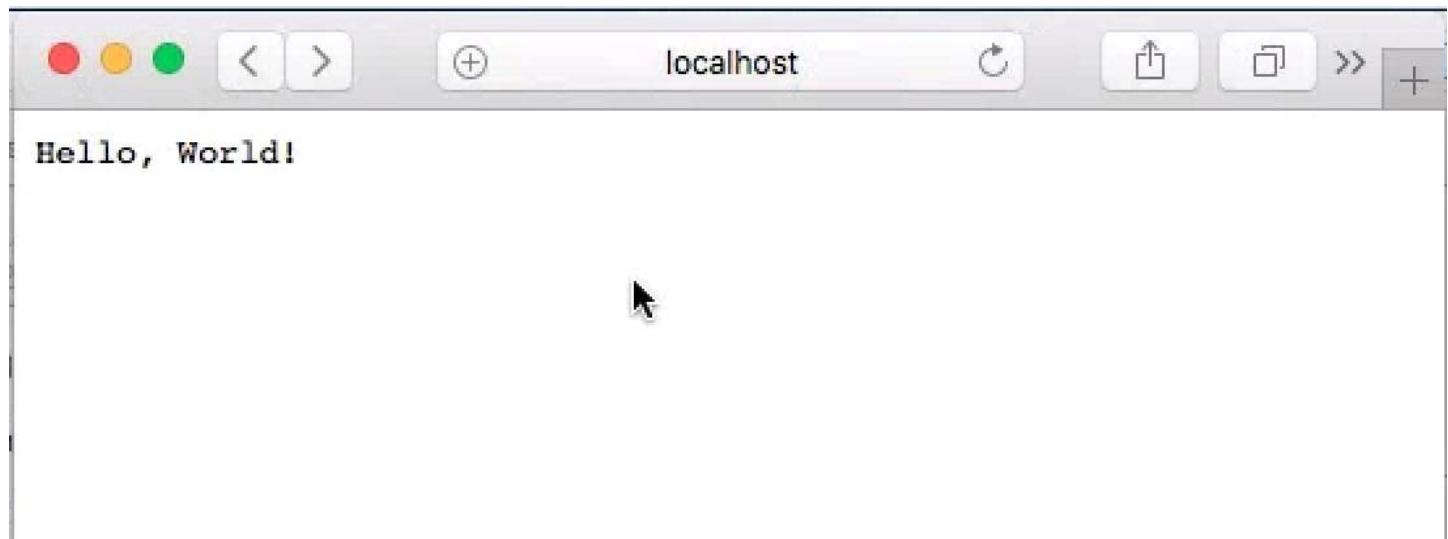
```
swift build
```

When the build is finished, executable will be placed at this location. Double click this executable to start the server.

```
.build/debug/kitura-helloworld
```

Validate

Open a browser, type in `localhost:8080/get` as url and hit enter. The hello world page should come out.



Open a HTTP request app, post plain text to `localhost:8080/post`. The respond string will show the entered text correctly.

localhost:8080/post

X

+

POST

localhost:8080/post

Authorization

Headers (1)

Body ●

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

Text ▾

1 Some text

I

Body

Cookies

Headers (4)

Tests

Pretty

Raw

Preview

Text ▾



1 Value Some text received.