



Martin J. Lasek



@martinlasek



@martin_lasek

I'm an always optimistic, open minded and knowledge seeking full-stack developer passionated about UI/UX and changing things for the better.



ServerSide.swift



Momoko IO



Martin J. Lasek



Vapor Developer



VaporBerlin



@martinlasek

The screenshot shows a dark-themed website for the ServerSide.swift conference. At the top right, there are navigation links for "HOME" (underlined), "SPEAKERS", and "SPONSORS". The main header features a large, stylized orange and red logo resembling a double-headed arrow or a lightning bolt. Below the logo, the text "ServerSide.swift" is displayed. The central section is titled "Server-Side Swift Conference" and "12th-14th September 2018. Berlin, Germany". A call-to-action section includes a text input field labeled "Email" and a red "Notify Me" button. The bottom of the page is divided into two columns: "The Conference" on the left and "Where and When" on the right. Both sections contain descriptive text and small images. At the very bottom, there are links for "Speakers" and "The Team".

HOME SPEAKERS SPONSORS

ServerSide.swift

Server-Side Swift Conference
12th-14th September 2018. Berlin, Germany

The best way to keep posted about the conference is by [signing up below!](#)
Next newsletter will contain everything about sponsor levels, ticket prices, schedule and more!

Email

Notify Me

The Conference

ServerSide.swift is a framework-independent conference, where we will learn and share on a number of different related topics. The conference is aimed at being a non-profit conference and solely run for the love of server-side Swift.

Where and When

The conference will take place in awesome Berlin, Germany and last for 3 days from 12th-14th September 2018!

Speakers

The Team

PERFECT

meets

KITURA

meets

VAPOR

Server-Side Swift Frameworks



PERFECT

KITURA

VAPOR

production ready

community

documentation

tutorial

books

Talk is cheap. Show
me the code.

Linus Torvalds

PERFECT

KITURA

VAPOR

```
~ git clone https://github.com/PerfectlySoft/PerfectTemplate.git helloPerfect  
~ cd helloPerfect  
~ swift package generate-xcodeproj
```

main.swift

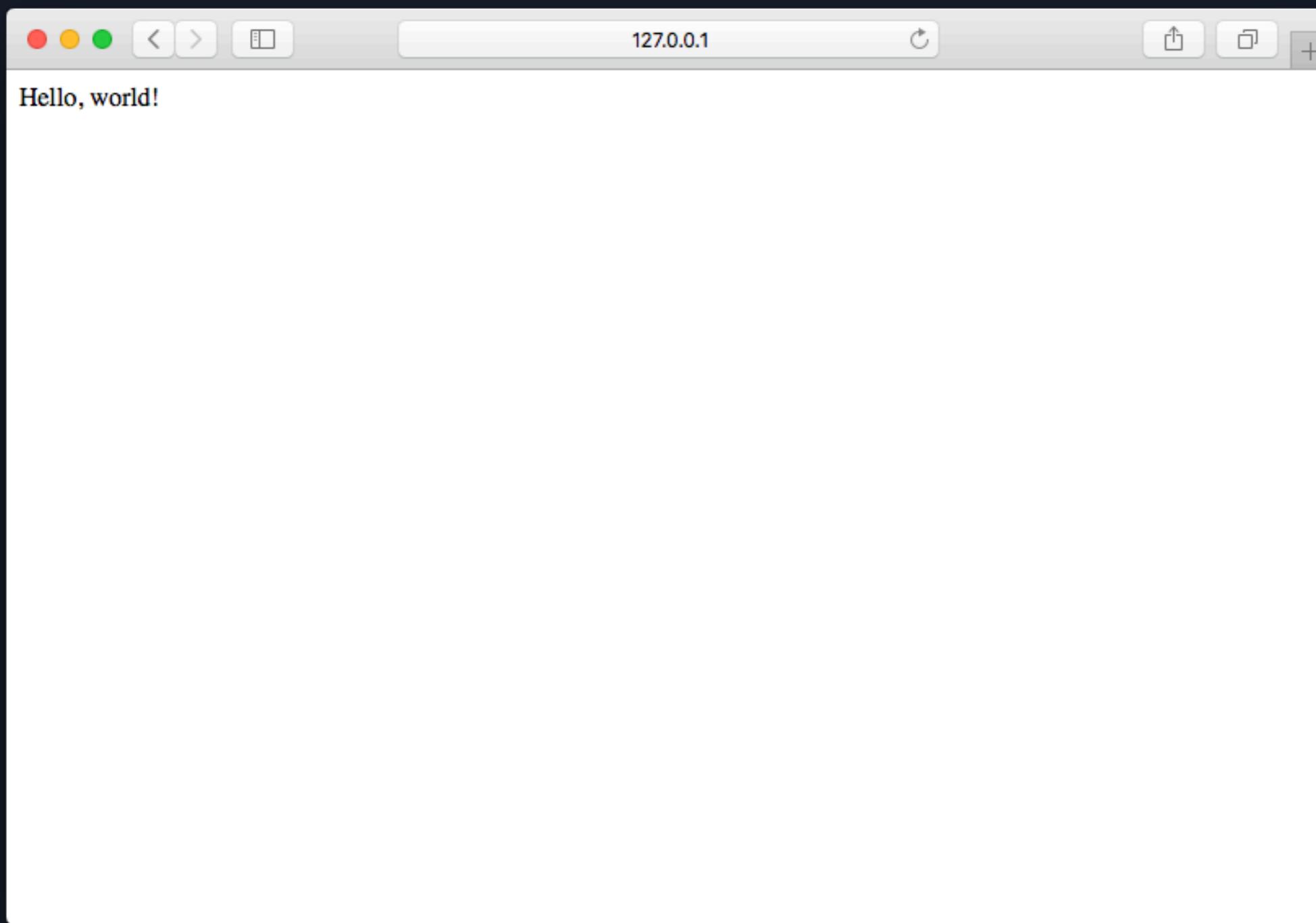
```
import PerfectHTTP  
import PerfectHTTPServer  
  
func handler(request: HTTPRequest, response: HTTPResponse) {  
    response.setHeader(.contentType, value: "text/html")  
    response.appendBody(string: "<html><title>Hello, world!</title><body>Hello, world!</body></html>")  
    response.completed()  
}  
  
let confData = [  
    "servers": [  
        [  
            "name": "localhost",  
            "port": 8181,  
            "routes": [  
                ["method": "get", "uri": "/", "handler": handler],  
                ["method": "get", "uri": "/**", "handler": PerfectHTTPServer.HTTPHandler.staticFiles,  
                 "documentRoot": "./webroot",  
                 "allowResponseFilters": true]  
            ],  
            "filters": [  
                [  
                    "type": "response",  
                    "priority": "high",  
                    "name": PerfectHTTPServer.HTTPFilter.contentCompression,  
                ]  
            ]  
        ]  
    ]  
]  
  
do {  
    try HTTPServer.launch(configurationData: confData)  
} catch {  
    fatalError("\(error)") // fatal error launching one of the servers  
}
```

PERFECT

KITURA

VAPOR

```
~ git clone https://github.com/PerfectlySoft/PerfectTemplate.git helloPerfect  
~ cd helloPerfect  
~ swift package generate-xcodeproj
```



PERFECT

KITURA

VAPOR

```
~ git clone https://github.com/PerfectlySoft/PerfectTemplate.git helloPerfect
~ cd helloPerfect
~ swift package generate-xcodeproj
```

main.swift

```
func handler(request: HTTPRequest, response: HTTPResponse) {
    ...
}

func json(request: HTTPRequest, response: HTTPResponse) {
    do {
        try response.setBody(json: ["conference": "ServerSide.swift"])
    } catch {
        /* But what if I don't want to.. */
    }
    response.completed()
}

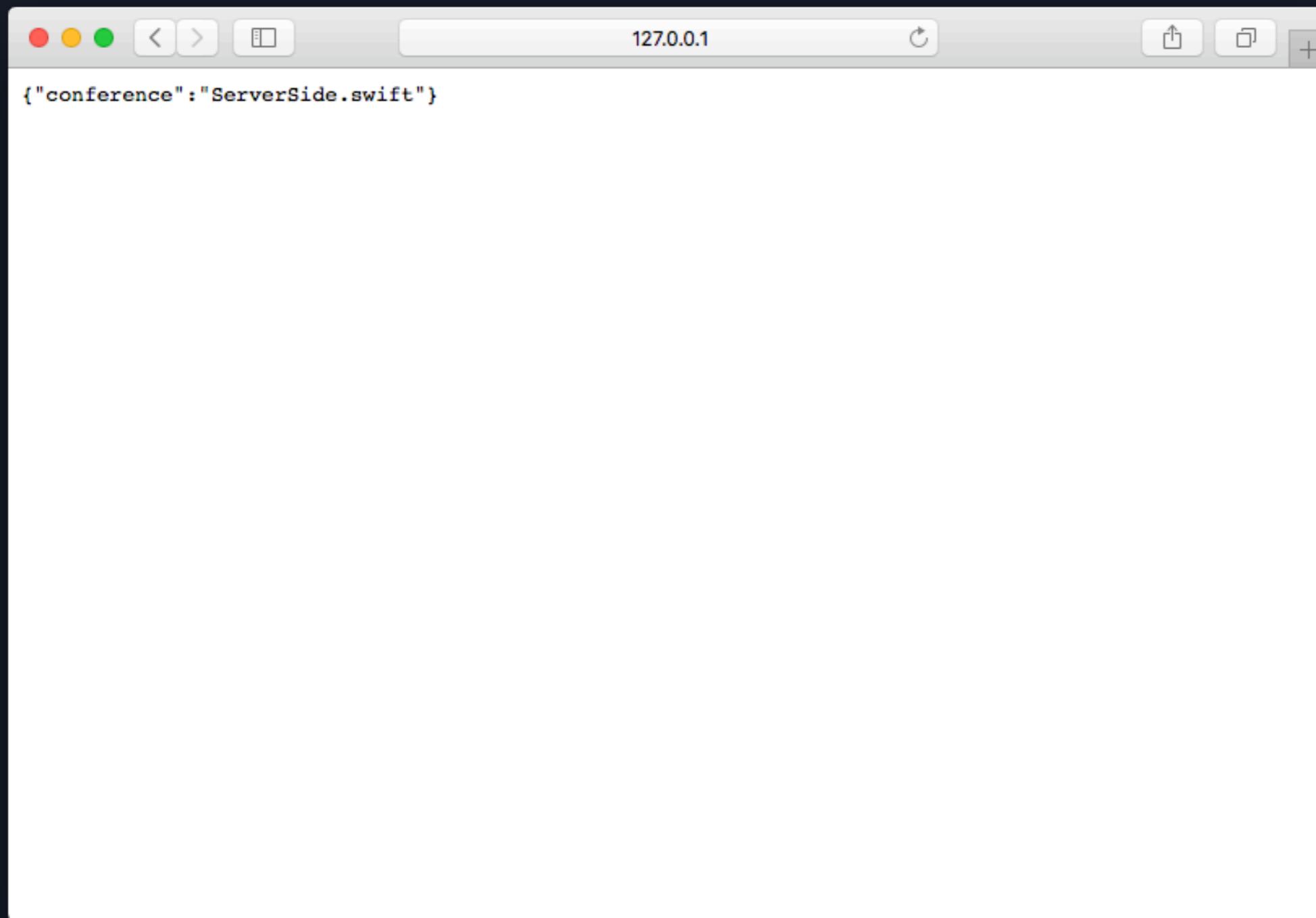
...
["method":"get", "uri":"/json", "handler": json],
...
```

PERFECT

KITURA

VAPOR

```
~ git clone https://github.com/PerfectlySoft/PerfectTemplate.git helloPerfect  
~ cd helloPerfect  
~ swift package generate-xcodeproj
```



PERFECT

KITURA

VAPOR

```
~ mkdir helloKitura  
~ cd helloKitura  
~ kitura init
```

HealthRoutes.swift

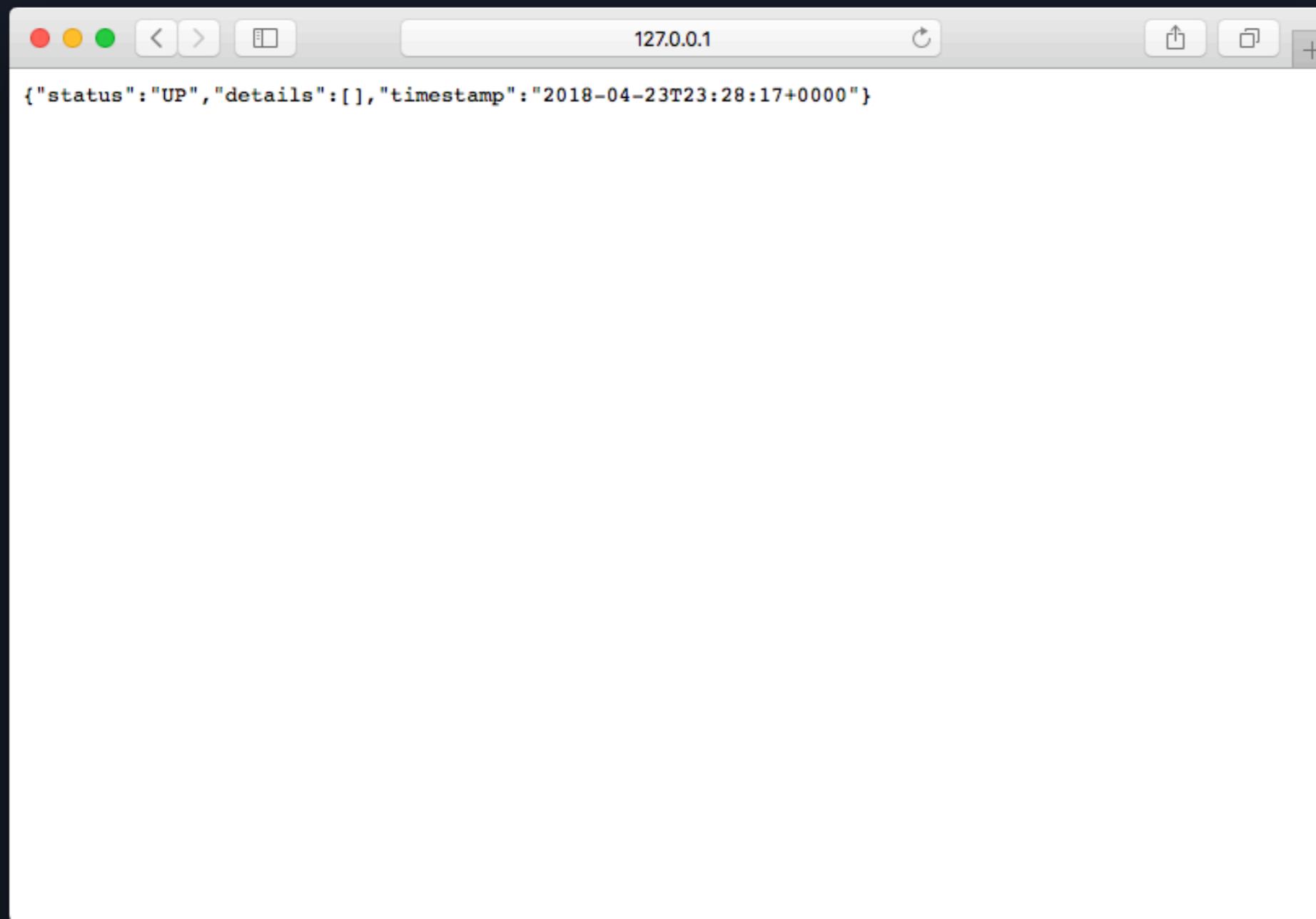
```
func initializeHealthRoutes(app: App) {  
  
    app.router.get("/health") { (respondWith: (Status?, RequestError?) -> Void) -> Void in  
        if health.status.state == .UP {  
            respondWith(health.status, nil)  
        } else {  
            respondWith(health.status, .serviceUnavailable)  
        }  
    }  
}
```

PERFECT

KITURA

VAPOR

```
~ mkdir helloKitura  
~ cd helloKitura  
~ kitura init
```



PERFECT

KITURA

VAPOR

```
~ mkdir helloKitura  
~ cd helloKitura  
~ kitura init
```

HealthRoutes.swift

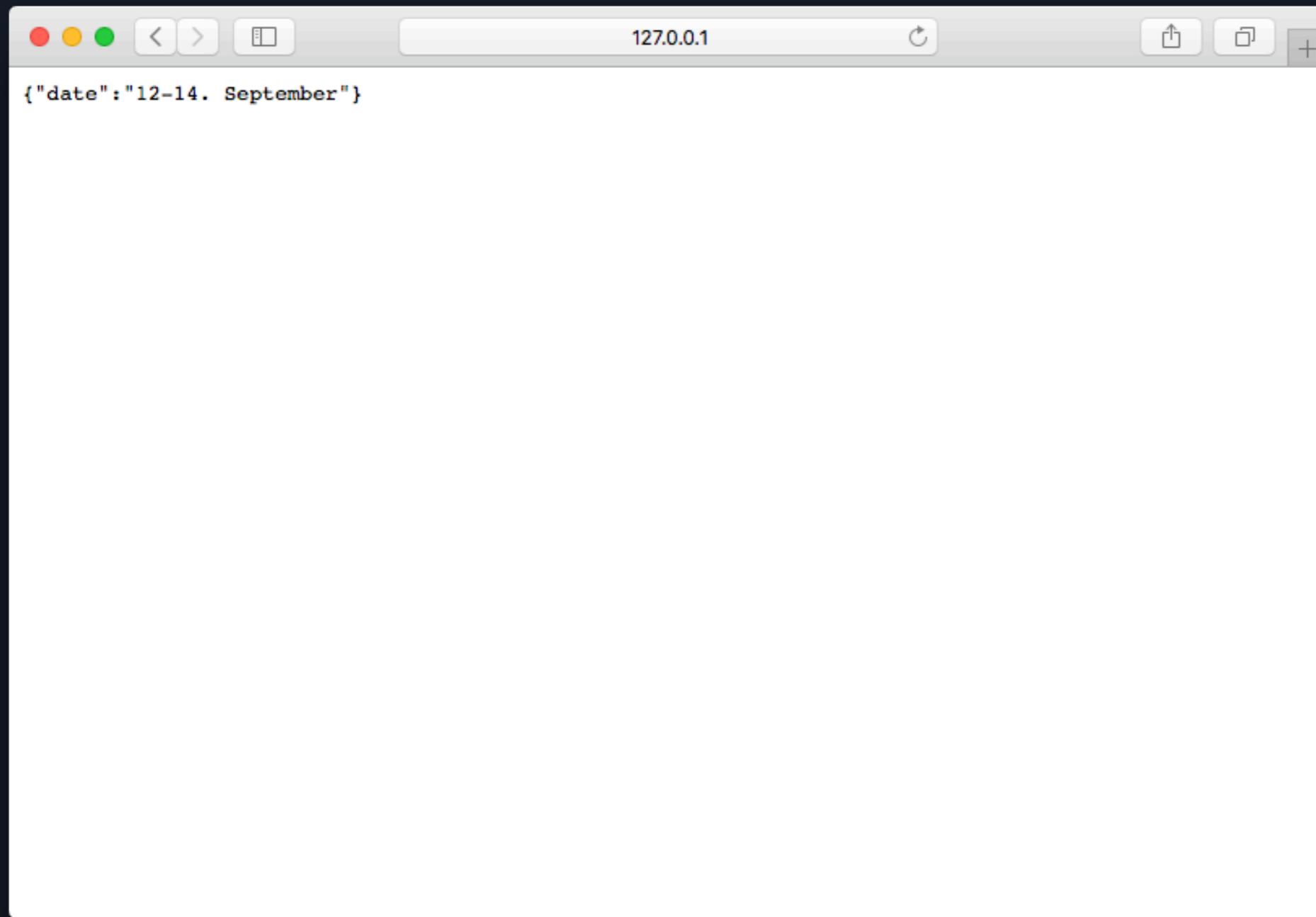
```
func initializeHealthRoutes(app: App) {  
  
    app.router.get("/health") { (respondWith: (Status?, RequestError?) -> Void) -> Void in  
        ...  
    }  
  
    app.router.get("/json") { request, response, _ in  
        response.status(.OK).send(json: ["date": "12-14. September"])  
    }  
}
```

PERFECT

KITURA

VAPOR

```
~ mkdir helloKitura  
~ cd helloKitura  
~ kitura init
```



PERFECT

KITURA

VAPOR

```
~ vapor new helloVapor --branch=beta  
~ cd helloVapor  
~ vapor xcode
```

routes.swift

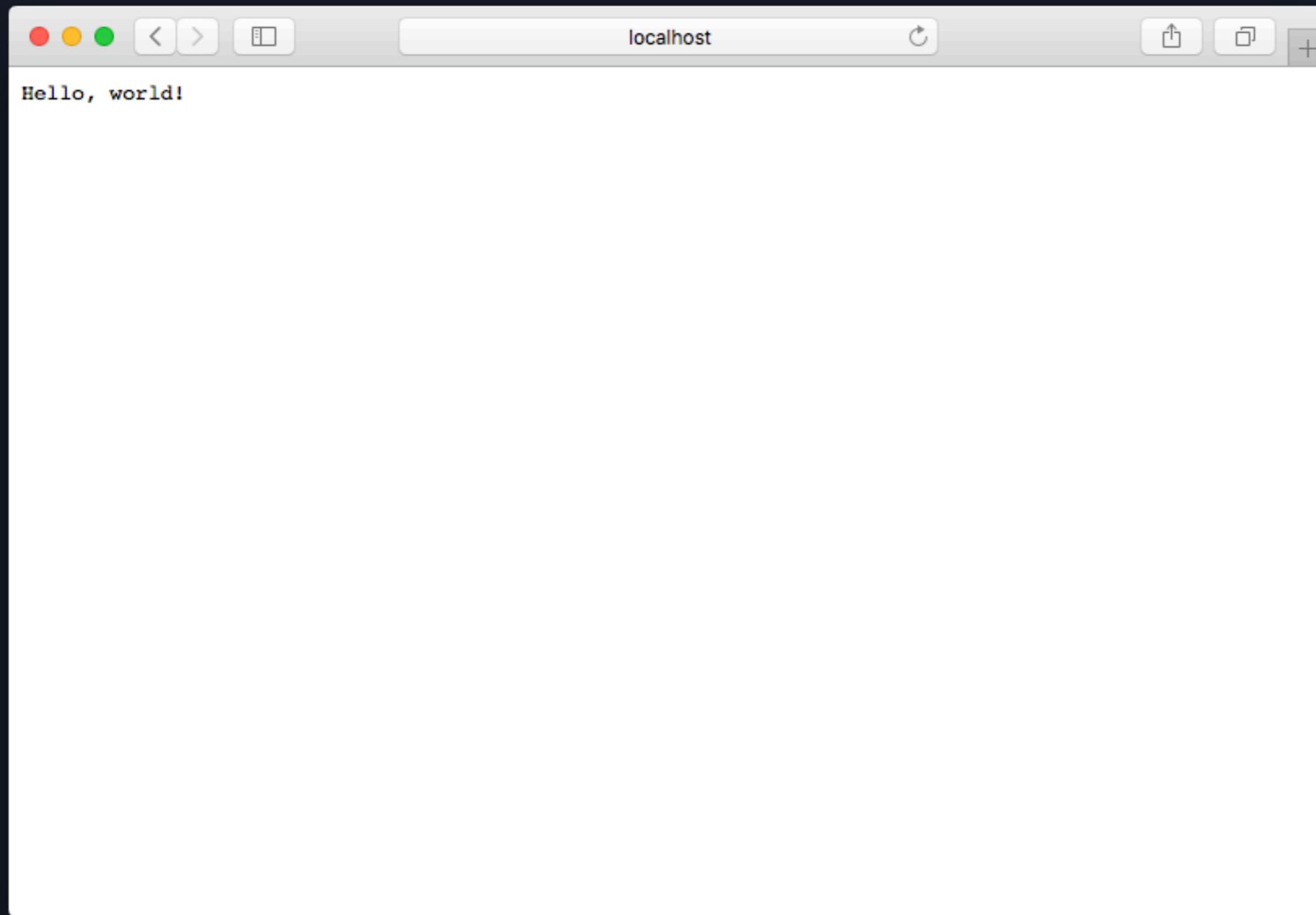
```
public func routes(_ router: Router) throws {  
    router.get("hello") { req in  
        return "Hello, world!"  
    }  
  
    let todoController = TodoController()  
    router.get("todos", use: todoController.index)  
    router.post("todos", use: todoController.create)  
    router.delete("todos", Todo.parameter, use: todoController.delete)  
}
```

PERFECT

KITURA

VAPOR

```
~ vapor new helloVapor --branch=beta  
~ cd helloVapor  
~ vapor xcode
```



PERFECT

KITURA

VAPOR

```
~ vapor new helloVapor --branch=beta  
~ cd helloVapor  
~ vapor xcode
```

routes.swift

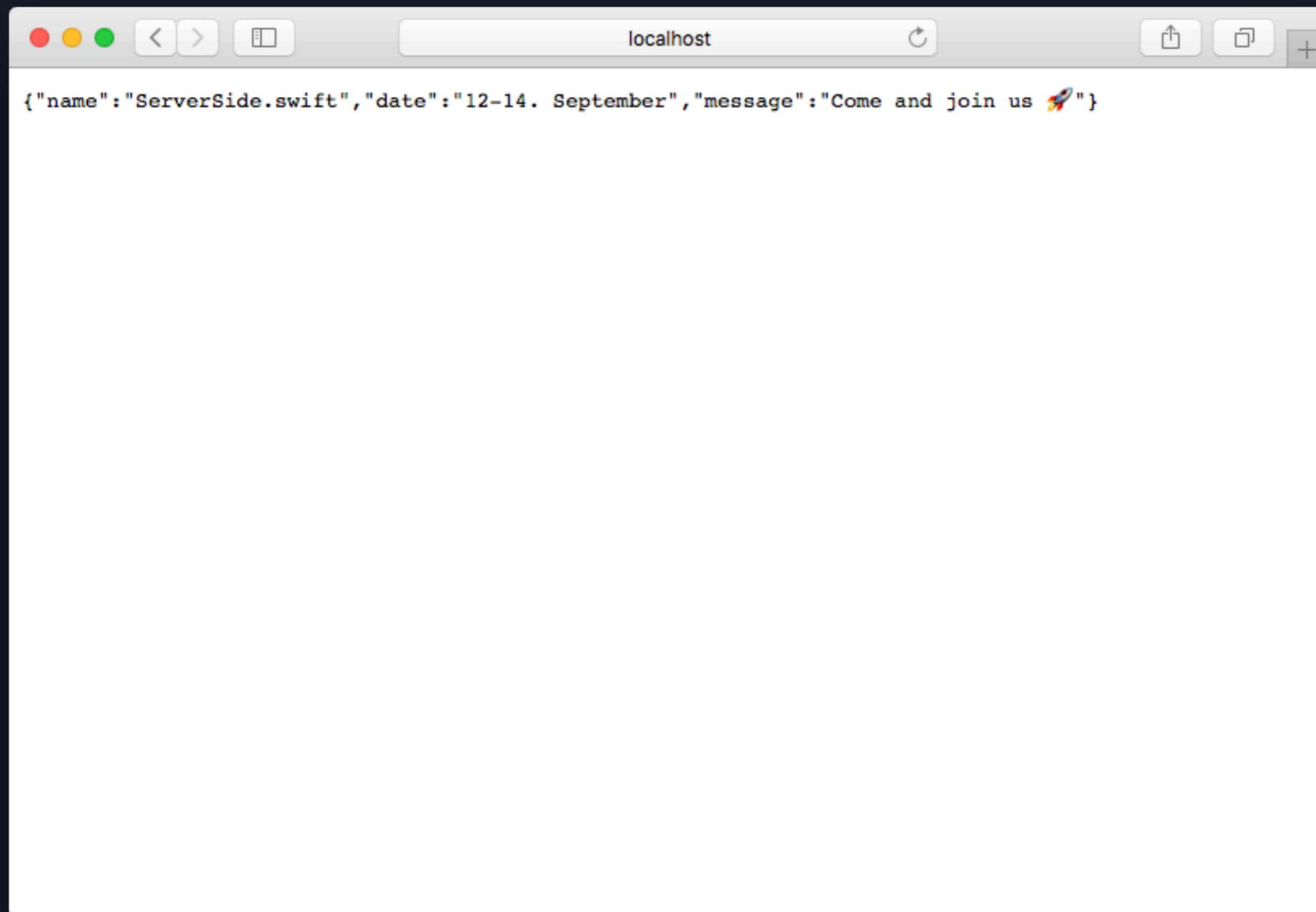
```
public func routes(_ router: Router) throws {  
  
    router.get("hello") { req in  
        return "Hello, world!"  
    }  
  
    router.get("json") { req in  
        return Conference(  
            name: "ServerSide.swift",  
            date: "12-14. September",  
            message: "Come and join us 🚀"  
        )  
    }  
    ...  
}  
  
struct Conference: Content {  
    var name: String  
    var date: String  
    var message: String  
}
```

PERFECT

KITURA

VAPOR

```
~ vapor new helloVapor --branch=beta  
~ cd helloVapor  
~ vapor xcode
```



PERFECT

www.perfect.org

KITURA

www.kitura.io

VAPOR

vapor.codes

