Ktor: Kotlin Backend Framework

Metas Kerdwat(Oak)

Growth Session #21 - January 24-25 2019

Intro

- Ktor is a Kotlin framework for building asynchronous servers
- Created by Jetbrains - the organization that created Kotlin
- It has a plugin for creating templates in Intellij
- It has a lot of build-in functionality such as Authorization, Data Parsing, or Error Handling

Ktor

Intro

Simplest example

```
fun main() {
    embeddedServer(factory = Netty, port = 8080) { this: Application
        routing { this: Routing
            get( path: "/") { this: PipelineContext<Unit, ApplicationCall>
                call.respondText( text: "Hello World", ContentType.Text.Plain)
            }
        }
    }.start(wait = true)
}
```

Setting up

```
fun Application.module() {
    install(DefaultHeaders)
    install(CallLogging)
    install(ContentNegotiation) { gson() }
    install(StatusPages) { /* Handling errors */ }
    install(Authentication) { /* Handling authentication */ }
    routing { /* Handling routing */ }
}
```

Intro

Many pluggable features

Error Handling

```
fun Application.installStatusPages() = install(StatusPages) { this: StatusPages.Configuration
    exception<AuthenticationException> { this: PipelineContext<Unit, ApplicationCall>
        call.respond(HttpStatusCode.Unauthorized, Message(it.toString()))
    }
    exception<AuthorizationException> { this: PipelineContext<Unit, ApplicationCall>
        call.respond(HttpStatusCode.Forbidden, Message(it.toString()))
    }
    exception<MissingParameterException> { this: PipelineContext<Unit, ApplicationCall>
        call.respond(HttpStatusCode.BadRequest, Message(it.msg))
    }
    exception<Throwable> { cause ->
        call.respond(HttpStatusCode.InternalServerError, Message( message: cause.message ?: cause.toString()))
    }
}
```

Authentication: Basic

```
fun Application.installAuthentication() = install(Authentication) { this: Authentication.Configuration
    basic( name: "BASIC_AUTH") { this: BasicAuthenticationProvider
        validate { credential ->
            if (credential.name == "admin" && credential.password == "root") {
                ^validate UserIdPrincipal(credential.name)
            } else {
                ^validate null
            }
        }
    }
}
```

Intro

It supports ORM

```
object Libraries : Table() {
    val id: Column<Int> = integer( name: "id").autoIncrement().primaryKey()
    val name: Column<String> = varchar( name: "name", length = 50).default( defaultValue: "")
    val version: Column<String> = varchar( name: "version", length = 20).default( defaultValue: "")
}
```

Example of performing a Select

```
fun selectAll(): List<LibraryItem> = Libraries
    .selectAll()
    .toList()
    .map { row ->
        LibraryItem(
            id = row[Libraries.id],
            name = row[Libraries.name],
            version = row[Libraries.version]
        )
    }
```

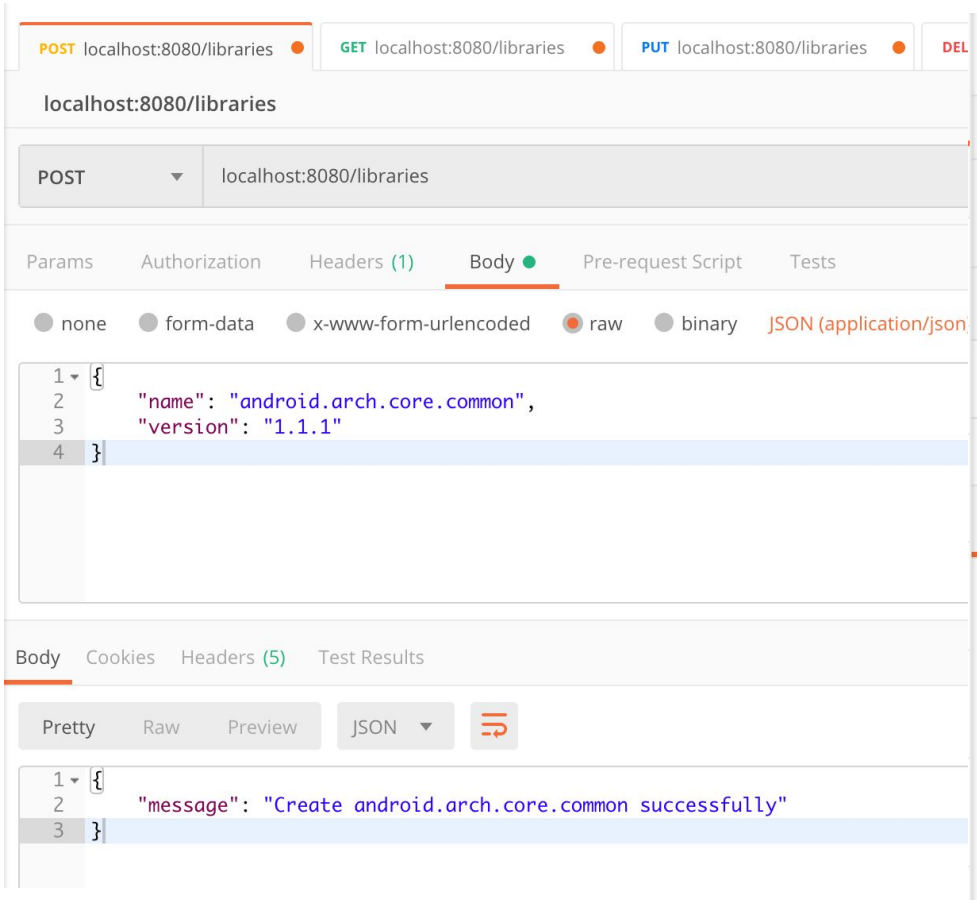
Goals

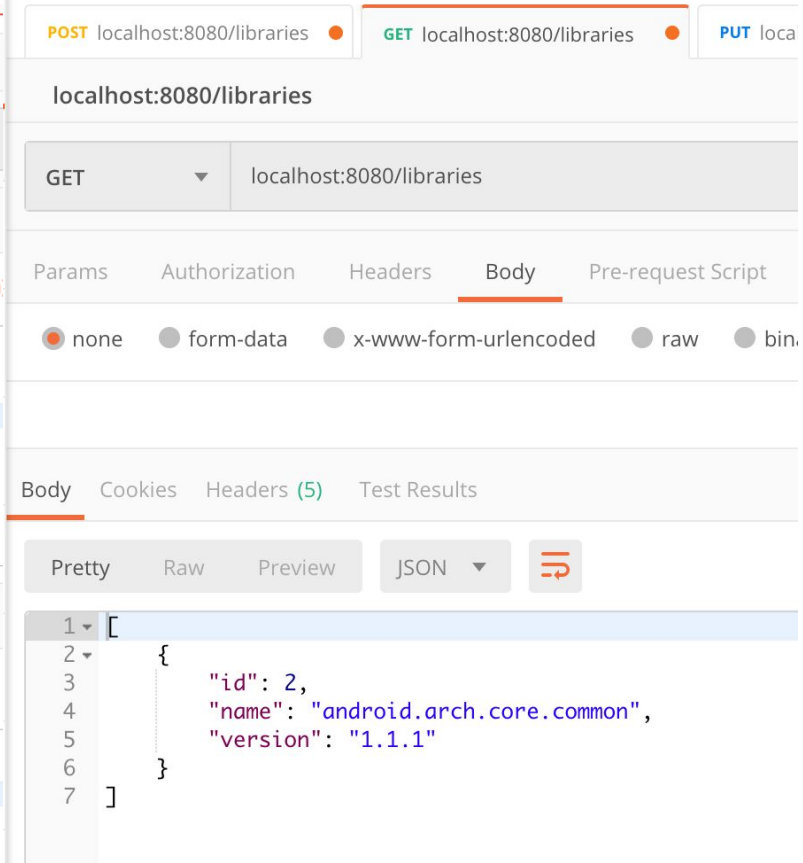
Create a CRUD web API for the **New Releases Notifier** made by Luong

Achievement and Progress

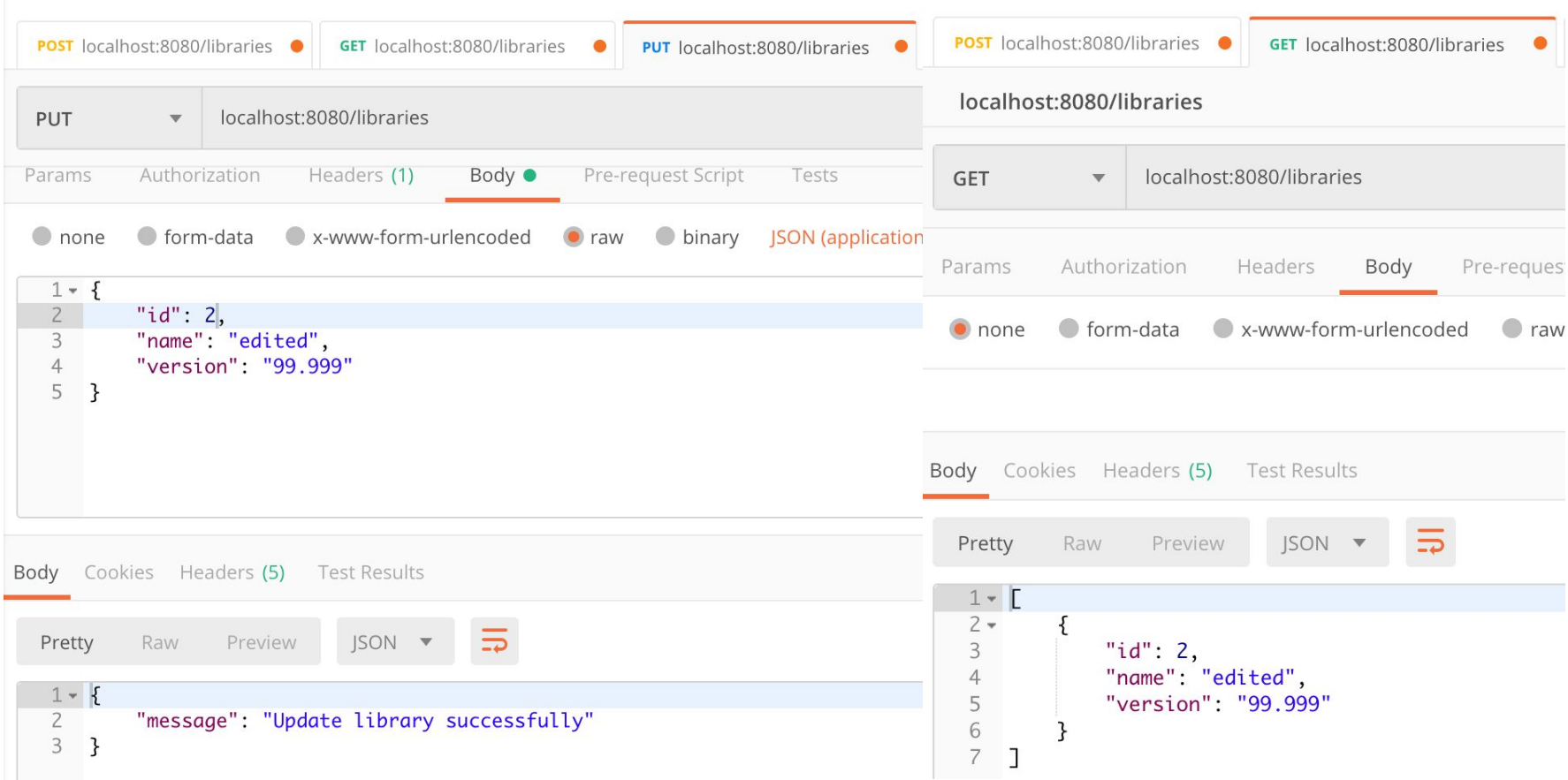
- Finish CRUD API
- Could not deploy to Heroku yet

Achievement and Progress

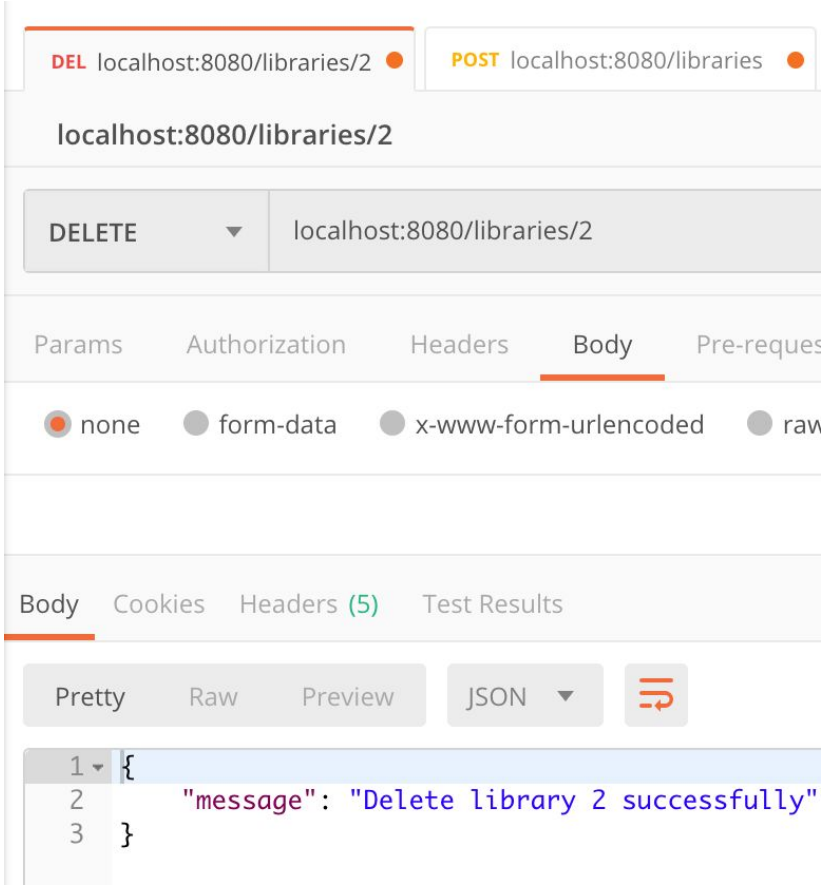


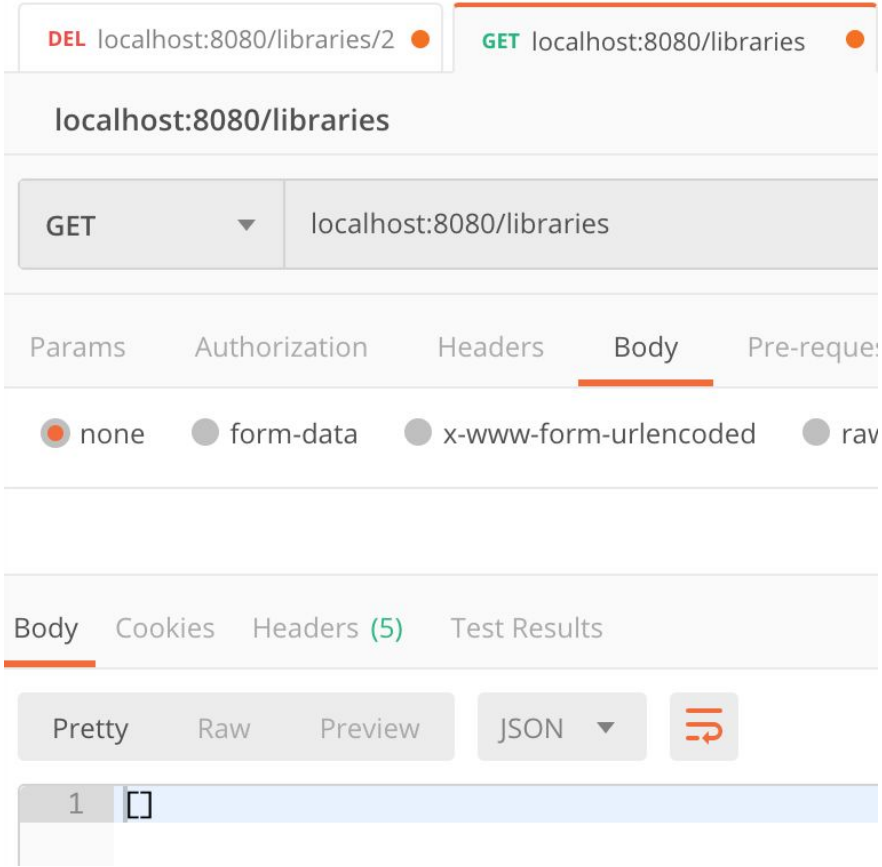


Achievement and Progress



Achievement and Progress





Conclusion

- Ktor is a backend framework that is suitable for those who have knowledge about the backend already
- The documentation is ok, could be better
- Kotlin knowledge is needed obviously

Next Steps

- Integrate backend service with Luong’s project if possible

Thanks!

Contact Nimble

nimblehq.co

hello@nimblehq.co

Bangkok

399 Interchange 21 Sukhumvit Road, Unit #2402-03, Klong Toei, Wattana, Bangkok 10110, Thailand

Singapore

28C Stanley St, Singapore 068737

Hong Kong

20th Floor, Central Tower

28 Queen's Road, Central, Hong Kong

nimble