Microservices

Webservices with Scala (II)

Content

Deep Dive into Play2

1. Database Access with Slick
2. Database Migration with Flyway
3. akka
    3.1. overview
    3.2. akka-http (the http server behind play)

Slick

Slick is a Scala library for working with relational databases. That means it allows you to model a schema, run queries, insert data, and update data[1]

- **Access databases using a similiar interface like collections (map, filter, flatMap, ...)**
- **Slick isn't an ORM (Object-Relational Mapping) - ORMs attempt to map object oriented data models onto relational database backends. By contrast, Slick provides a more database-like set of tools such as queries, rows and columns.**

[1] Comparing Scala relational database access libraries

Slick

- First you have to define your Schema and your Table (in our exercise it's automatically generated for you)
- Then you can do Queries or do other tasks on your data

```
// schema & table
final case class Message(sender: String, content: String, id: Long = 0L)

final class MessageTable(tag: Tag) extends Table[Message](tag, "message") {
    def id = column[Long]("id", O.PrimaryKey, O.AutoInc)
    def sender = column[String]("sender")
    def content = column[String]("content")
    def * = (sender, content, id).mapTo[Message]
}

// data to insert
def freshTestData = Seq(
    Message("Dave", "Hello, HAL. Do you read me, HAL?"),
    Message("HAL", "Affirmative, Dave. I read you."),
    Message("Dave", "Open the pod bay doors, HAL."),
    Message("HAL", "I'm sorry, Dave. I'm afraid I can't do that.")
)
val insert: DBIO[Option[Int]] = messages ++= freshTestData
val result: Future[Option[Int]] = db.run(insert)

// selecting data
lazy val messages = TableQuery[MessageTable]
messages.filter(_.sender === "HAL")
// more about slick 3: https://books.underscore.io/essential-slick/essential-slick-3.pdf
```

Flyway

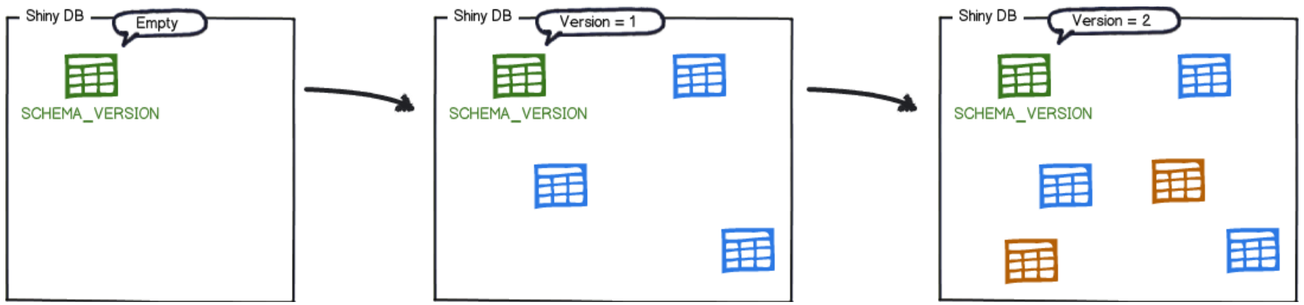
Flyway: Version control for your database. Robust schema evolution across all your environments. With ease, pleasure and plain SQL.

Very good for Continuous Database Integration: Continuous Database Integration (CDBI) is the process of rebuilding your database and test data any time a change is applied to a project's version control repository (later @ DevOps)

What is exactly Flyway?

- database migration framework for relational databases based on Java
- build a database from scratch
- manages the database and schema
- mulitple modes for migrating

  `V1__Initial_Setup.sql, V2__First_Changes.sql`



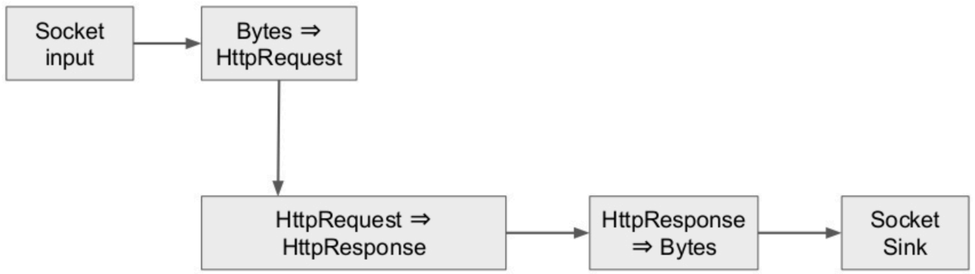
akka

akka is a toolkit for building highly concurrent, distributed and fault tolerant applications based on the actor model which runs on the JVM. Scale UP and OUT for free, because it's distributable by design.

- **For distribution across threads --> Actors (perhaps with persistence and event sourcing)**
- **For distribution across machines --> akka clusters and akka remote**
- **interact with external world --> akka http, akka gRPC and alpakka**

akka-http

- **It is a toolkit based on akka streams and follows the reactive streams idea** `HTTPServer as Flow[HttpRequest, HttpResponse]`
- **It is not a web framework (look @ play2)**
- **It has full server and client-side HTTP stack**
- **It has a mulitple level API**



akka-http implementation

- APIs in both Scala and Java
- Fully asynchronous and nonblocking
- Focused on higher level API
- lightweight & modular
- testable

```
def routeGetOne =
 get {
  path(IntNumber) { dummyId =>
   authenticate { u =>
    complete {
     dummyService.getOne(dummyId).map(_.asJson)
    }
   }
  }
 }
//scalatest
Get("/1") ~> routeGetOne ~> check {
    status == OK
    entity.as[Dummy] === ???
}


//out of: https://github.com/innFactory/bootstrap-akka-http
```