

SCALA SERVER TOOLKIT

HOW TO WRITE FP MICROSERVICE STEP-BY-STEP

Jakub Janeček, Avast Software
LambdaConf 2020 Global Edition
June 23, 2020



SCA.LA/SERVER-TOOLKIT

A BIT OF HISTORY

- » Scala at Avast since around 2011.
- » Better Java → idiomatic Scala → FP.
- » Diverse codebases lacking unified approach.
- » Lots of in-house solutions.

ISSUES

- » Lack of common style.
- » Hard to get newcomers up to speed.
- » Maintenance of internal libraries.

"SOLUTION"

- » Focus on open source.
- » Unify the way components/libraries are initialized.
- » Provide missing integration pieces.
- »  Scala Server Toolkit

ZIO SIDENOTE

- » ZLayer should have been available before!
- » SST would probably not emerge.
- » On the other hand:
 - » Constructor-based injection is more approachable to newcomers.
 - » Useful to people not wanting to go full ZIO.

DESIGN

- » Modular based on dependencies.
- » Functional programming.
- » Type safe configuration.
- » Proper resource management.
- » No DI framework - just constructors.

WHAT IS AVAILABLE?

- » http4s server/client (Blaze)
- » JVM - console, random, thread pools
- » doobie (JDBC)
- » Cassandra (official Datastax driver)
- » Flyway (DB migrations)
- » Micrometer - JMX, StatsD
- » PureConfig
- » Bundles for ZIO and Monix

MODULE

```
object ComponentModule {  
    def make[F[_]](config: ComponentConfig, someDependency: Dependency): Resource[F, Component] = {  
        ???  
    }  
}
```

MODULE COMPOSITION

```
for {  
    a ← ComponentAModule.make[Task]  
    b ← ComponentBModule.make[Task](a)  
    c ← ComponentCModule.make[Task](a, b)  
} yield a
```



Q&A

Thank you.

- » <https://github.com/avast/scala-server-toolkit>
- » <https://github.com/jakubjanecek>
- » @jakubjanecek