Kotlin+MicroProfile: Teaching 20 year old tricks to a new language

Víctor Orozco - Nabenik

September 10, 2019

@tuxtor

Java - Java as in JVM

TheServerSide
Your Enterprise Java Community

Java management ▼ | All Subtopics | Search TechTa

Transcipt of James' TSSJS 2011 Discussion about Java and the JVM

"At the core of the Java ecosystem is the JVM. Most people talk about Java the language, and this may sound odd coming from me, but I could hardly care less.

"What I really care about is the Java Virtual Machine as a concept, because that is the thing that ties it all together; it's

> Most people talk about Java the language, and this may sound odd coming from me, but I could hardly care less.
>
> James Gosling

Microservices

Microservices

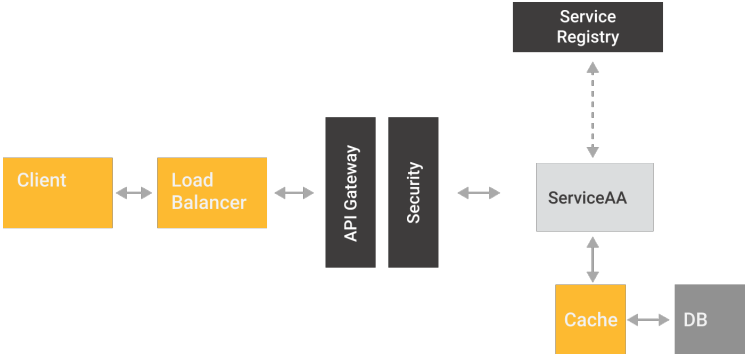


Figure 1: Microservices

12 cloud native factors (Heroku)

MICROPROFILE
OPTIMIZING ENTERPRISE JAVA

Microprofile

- Config
- Backing service
- Processes (Stateless REST)
- Disposability (Fail with style)

Cloud

- Codebase (Git-Flow)
- Dependencies (Maven)
- Build, Release, Run (Pipelines)
- Port binding
- Concurrency (Docker - k8s)
- Dev / Prod parity
- Logs
- Admin process

Microservices - Java

- DIY - Jooby, Javalin, Micronaut, Spark, Vert.x, Helidon SE
- Enterprise - Spring Boot, Eclipse MicroProfile

Microservices - Kotlin

- DIY - Jooby, Javalin, Micronaut, Spark, Vert.x, Helidon SE, **Ktor**
- Enterprise - Spring Boot, Eclipse MicroProfile

Microservices - Why Jakarta EE with Kotlin?

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

- Jakarta EE standards are pervasive
- In LATAM is easier to find EE developers
- Not every piece of software should be over-engineered
- Enterprises like "boring" and "old" software stacks, me too
- Is possible to create "fresh" software with already running app servers

Eclipse MicroProfile

Eclipse MicroProfile

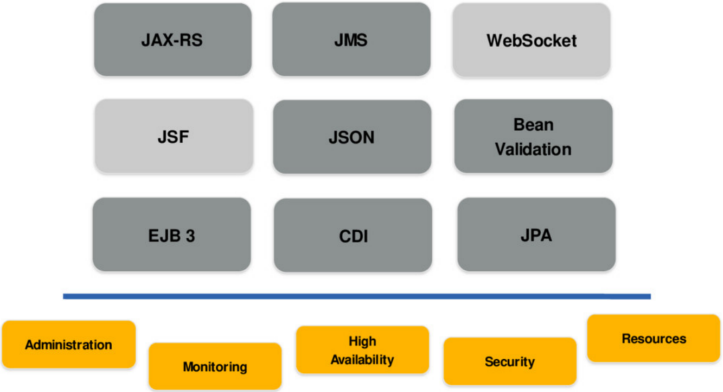
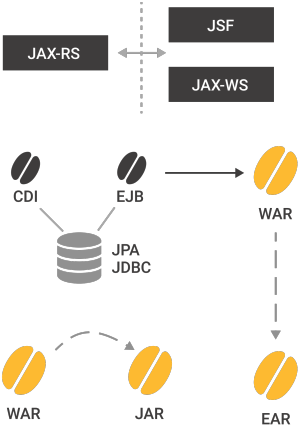


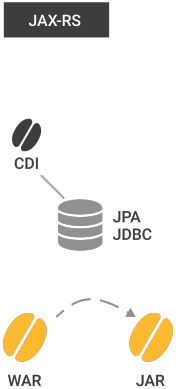
Figure 2: Credits: Reza Rahman

Eclipse MicroProfile



BV + JSON-P JSON-B + Security

Eclipse MicroProfile



BV + JSON-P JSON-B + Security

Eclipse MicroProfile

Eclipse MicroProfile - Ecosystem

Libraries

- SmallRye
- Hammock
- Apache Geronimo
- Fujitsu Launcher

JEAS - Fat Jar, Uber Jar

- Dropwizard
- KumuluzEE
- Helidon (Oracle)
- Open Liberty (IBM)
- Apache Meecrowave
- Thorntail/Quarkus (Red Hat)
- Payara Micro

Eclipse MicroProfile - Ecosystem

Micro server - Thin War

- Payara Micro
- TomEE JAX-RS

Full server

- Payara Application Server
- JBoss Application Server / Wildfly Application Server
- WebSphere Liberty (IBM)

https://wiki.eclipse.org/MicroProfile/Implementation

Eclipse MicroProfile - 1, 2, 3 with Kotlin

1. Maven or Gradle config
2. MicroProfile dependency and your extras (Jakarta EE, Arquillian, JUnit, . . .)
3. Maven plugin (maven-compiler-plugin)
4. Kotlin plugin (kotlin-maven-plugin)

Eclipse MicroProfile + Kotlin + Maven

Eclipse MicroProfile with Payara 5

```
<dependency>
        <groupId>org.eclipse.microprofile</groupId>
        <artifactId>microprofile</artifactId>
        <type>pom</type>
        <version>2.1</version>
        <scope>provided</scope>
</dependency>
```

Kotlin with Maven - Dependency

```
<dependency>
        <groupId>org.jetbrains.kotlin</groupId>
        <artifactId>kotlin-stdlib-jdk8</artifactId>
        <version>${kotlin.version}</version>
</dependency>
```

Kotlin with Maven - maven-compiler-plugin

```
<execution>
        <id>default-compile</id>
        <phase>none</phase>
</execution>
<execution>
        <id>default-testCompile</id>
        <phase>none</phase>
</execution>
<execution>
        <id>java-compile</id>
        <phase>compile</phase>
        <goals> <goal>compile</goal> </goals>
</execution>
<execution>
        <id>java-test-compile</id>
        <phase>test-compile</phase>
        <goals> <goal>testCompile</goal> </goals>
</execution>
```

Kotlin with Maven - kotlin-maven-plugin

```
<compilerPlugins>
<plugin>all-open</plugin>
</compilerPlugins>
...
<option>all-open:annotation=javax.ws.rs.Path</option>
<option>all-open:annotation=javax.enterprise.context.RequestScoped</option>
<option>all-open:annotation=javax.enterprise.context.SessionScoped</option>
<option>all-open:annotation=javax.enterprise.context.ApplicationScoped</option>
<option>all-open:annotation=javax.enterprise.context.Dependent</option>
<option>all-open:annotation=javax.ejb.Singleton</option>
<option>all-open:annotation=javax.ejb.Stateful</option>
<option>all-open:annotation=javax.ejb.Stateless</option>
```

General idea: Just add all architectural annotations (CDI and EJB)

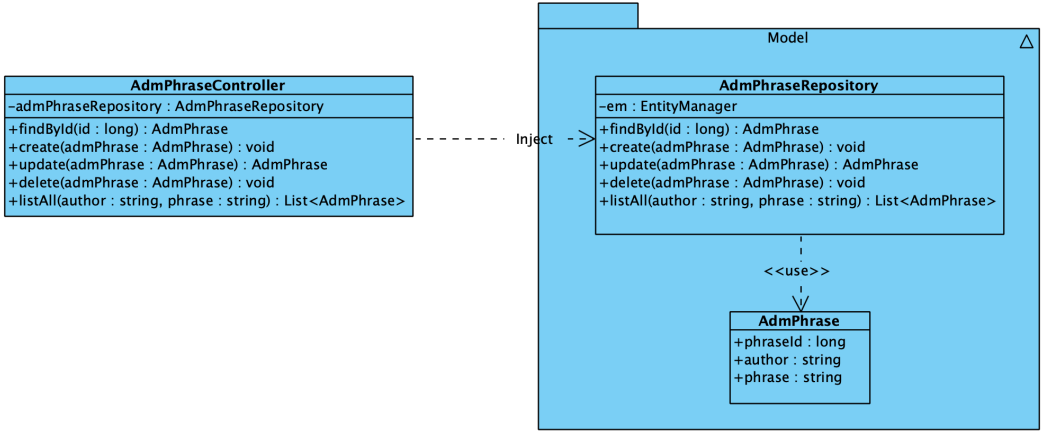
Demo

Kotlin + Jakarta EE + MicroProfile - Demo

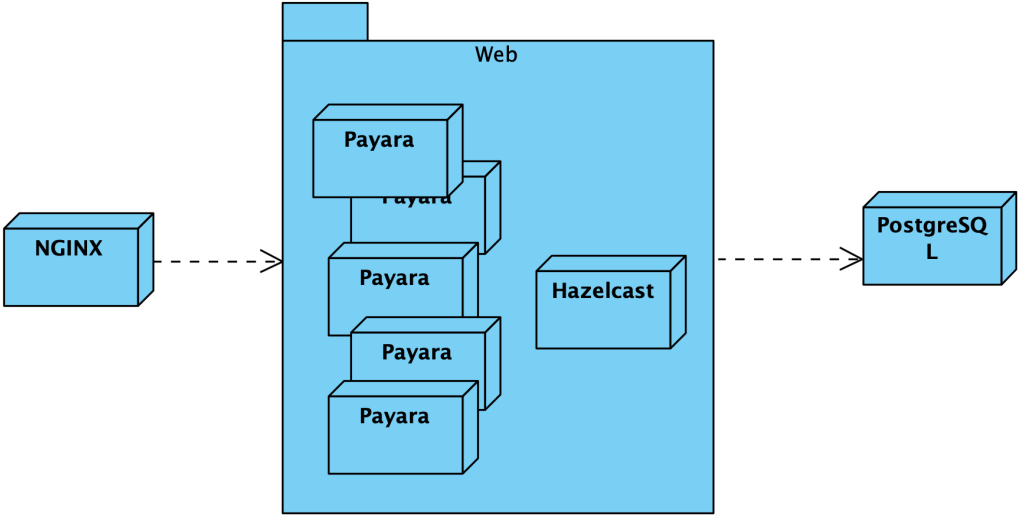
- Kotlin 1.3
- Libraries - SLF4J, Flyway, PostgreSQL
- Jakarta EE 8 - EJB, JPA
- MicroProfile - CDI, JAX-RS, MicroProfile Config
- Testing - Arquillian, JUnit, Payara Embedded

https://dzone.com/articles/
the-state-of-kotlin-for-jakarta-eemicroprofile-tra
https://github.com/tuxtor/integrum-ee

Kotlin + Jakarta EE + MicroProfile - Demo



Kotlin + Jakarta EE + MicroProfile - Demo



Kotlin - JPA entity

```kotlin
@Entity
@Table(name = "adm_phrase")
@TableGenerator(...)
data class AdmPhrase(
        @Id
        @GeneratedValue(strategy = GenerationType.TABLE,
                generator = "admPhraseIdGenerator")
        @Column(name = "phrase_id")
        var phraseId:Long? = null,
        var author:String = "",
        var phrase:String = ""
)
```

Data Clases, Nullable Types

Kotlin - CDI Repository

```kotlin
@RequestScoped
class AdmPhraseRepository {

        @Inject
        private lateinit var em:EntityManager

        ...

}
```

Lateinit (nullable type)

Kotlin - CDI Repository

```
fun create(admPhrase:AdmPhrase) = em.persist(admPhrase)

fun update(admPhrase:AdmPhrase) = em.merge(admPhrase)

fun findById(phraseId: Long) =
em.find(AdmPhrase::class.java, phraseId)

fun delete(admPhrase: AdmPhrase) = em.remove(admPhrase)
. . .
```

Single expression functions (One line methods)

Kotlin - CDI Repository

```
fun listAll(author: String, phrase: String):
        List<AdmPhrase> {

        val query = """SELECT␣p␣FROM␣AdmPhrase␣p
␣␣␣␣␣␣␣␣where␣p.author␣LIKE␣:author
␣␣␣␣␣␣␣␣and␣p.phrase␣LIKE␣:phrase
␣␣␣␣␣␣␣␣"""

        return em.createQuery(query, AdmPhrase::class.java)
                .setParameter("author", "%$author%")
                .setParameter("phrase", "%$phrase%")
                .resultList
}
```

Multiline string

Kotlin - JAX-RS Controllers

```kotlin
@Path("/phrases")
@Produces(MediaType.APPLICATION_JSON)
@Consumes(MediaType.APPLICATION_JSON)
class AdmPhraseController{

        @Inject
        private lateinit var admPhraseRepository: AdmPhraseRepository

        @Inject
        private lateinit var logger: Logger
        ...

}
```

Kotlin - JAX-RS Controller

```
@GET
fun findAll(
@QueryParam("author") @DefaultValue("%") author: String,
@QueryParam("phrase") @DefaultValue("%") phrase: String) =
        admPhraseRepository.listAll(author, phrase)

@GET
@Path("/{id:[0-9][0-9]*}")
fun findById(@PathParam("id") id:Long) =
        admPhraseRepository.findById(id)

@PUT
fun create(phrase: AdmPhrase): Response {
        admPhraseRepository.create(phrase)
        return Response.ok().build()
}
```

Kotlin - JAX-RS Controller

Elvis operator as expression

```
@POST
@Path("/{id:[0-9][0-9]*}")
fun update(@PathParam("id") id: Long?, phrase: AdmPhrase)
        :Response {
        if(id != phrase.phraseId)
                return Response.status(Response.Status.NOT_FOUND).build()

        val updatedEntity = admPhraseRepository.update(phrase)
        return Response.ok(updatedEntity).build()
}

@DELETE
@Path("/{id:[0-9][0-9]*}")
fun delete(@PathParam("id") id: Long): Response {
        val updatedEntity = admPhraseRepository.findById(id) ?:
        return Response.status(Response.Status.NOT_FOUND).build()
        admPhraseRepository.delete(updatedEntity)
        return Response.ok().build()
```

Oracle Cloud

```
<groupId>io.fabric8</groupId>
<artifactId>docker-maven-plugin</artifactId>
<version>0.30.0</version>
...
<image>
        <name>iad.ocir.io/tuxtor/microprofile/integrum-ee</name>
        <build>
                <dockerFile>${project.basedir}/Dockerfile</dockerFile >
        </build>
</image>
```

Oracle Cloud

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

Registry

Create Repository

tuxtor

- microprofile (Public)
- microprofile/hello-ee
- microprofile/hello-escalable
- microprofile/home-ee
- microprofile/integrum-ee
  - 1
  - latest
- microprofile/jvmservice
- microprofile/omdb-demo
- microprofile/payara-demo (Public)

microprofile/integrum-ee

User: ...42db3y5hmh4zajq Show Copy

Created: a month ago

Access: Private

Size: 138.19 MB

Last Push: 39 minutes ago

Readme

No readme has been created yet for this repository.

Oracle Cloud

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

MENU

ORACLE®
Cloud Infrastructure

Compute » Instances » Instance Details

RUNNING

instance-20181206-0243

Create Custom Image | Start | Stop | Reboot | Terminate | Apply Tag(s)

Instance Information | Tags

Instance Information

Availability Domain: hgWe:US-ASHBURN-AD-1

Fault Domain: FAULT-DOMAIN-2

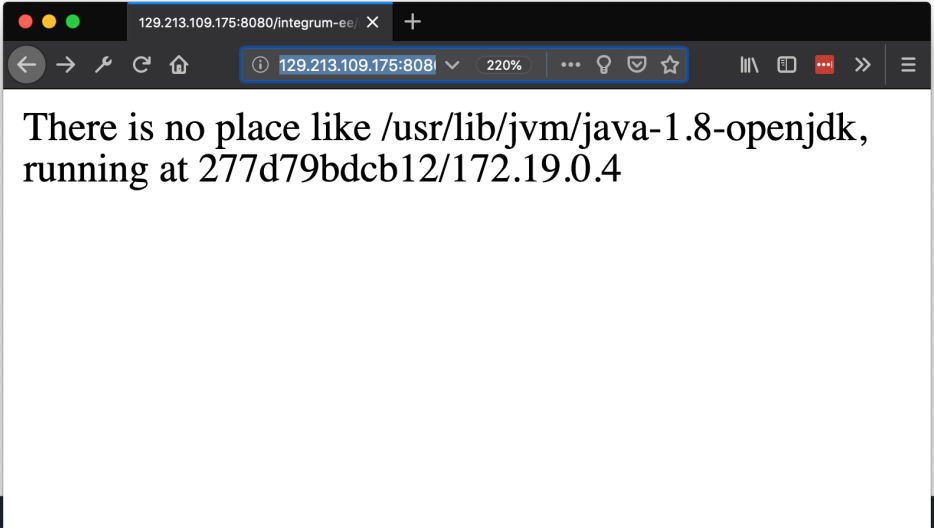
Region: iad

Shape: VM.Standard2.1

Oracle Cloud

1. bash

```
tuxtor@millenium-falcon-2:~/GitHub/integrum-ee$ mvn docker:build
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building integrum-ee 2.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- docker-maven-plugin:0.30.0:build (default-cli) @ integrum-ee ---
[INFO] Building tar: /Users/tuxtor/GitHub/integrum-ee/target/docker/iad.ocir.io/tuxtor/microprofile/integrum-ee/t
mp/docker-build.tar
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Created docker-build.tar in 145 milliseconds
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Built image sha256:26156
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Removed old image sha256:a2361
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.765 s
[INFO] Finished at: 2019-05-30T16:39:15-06:00
[INFO] Final Memory: 17M/239M
[INFO] ------------------------------------------------------------------------
tuxtor@millenium-falcon-2:~/GitHub/integrum-ee$ mvn docker:push
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building integrum-ee 2.0-SNAPSHOT
```


129.213.109.175:8080/integrum-ee/
129.213.109.175:808
220%
There is no place like /usr/lib/jvm/java-1.8-openjdk, running at 277d79bdcb12/172.19.0.4

Kotlin

Kotlin

- Static typing
- Java inter-op
- OO + FP
- Null safety
- Extension functions
- Operator overloading
- Data classes
- One line methods

Kotlin - Interesting facts

- Effective Java - Immutability, builder, singleton, override, final by default, variance by generics
- Elvis - Groovy
- Type inference - Scala
- Immutability - Scala
- Identifiers - Scala
- Null values management - Groovy
- Functions - Groovy

Java - Dying since 1995

- Spring Boot, Micronaut, MicroProfile, GraalVM . . .
- Raw performance (Beam, Spark, Hadoop)
- Tooling - IDE, Maven, Drivers RDBMS
- JVM - (Twitter, Alibaba, Spotify, etc.)
- OpenJDK

Kotlin

Advantages

- Concise code once you get the new structures
- Good Java inter-op
- Opening backend for new Android devs
- A new "Full-stack" approach

Disadvantages

- IntelliJ IDEA Ultimate
- Steep learning curve
- Compiler (time)
- Thread-managed vs Co-routines
- Amber, Loom, Valhalla, Panama (Java 16?)

- vorozco@nabenik.com
- @tuxtor
- https://vorozco.com

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0.