



Quarkus World Tour @Consol: Messaging

Dr. Marco Bungart
Senior Software Engineer

2021-07-01



```
$> whoami
```

Dr. Marco Bungart

- 2008 – 2013: Studium Bioinformatik/Informatik in Jena
- 2013 – 2018: Promotion in Kassel
- Seit 2018: Software Engineer at ConSol
- Twitter, github, bitbucket, stackoverflow, ... : turing85
- Interessenschwerpunkte: Keycloak, GraalVM, Quarkus





Messaging über MicroProfile

Messaging über MicroProfile

- Microprofile-Standard → Wechseln von z.B. JMS auf Kafka ist (von Codeseite her) sehr einfach
- Asynchrone Kommunikation
- Einseitige Kommunikation
- Best Practice: synchrone und asynchrone Kommunikation nicht vermischen (z.B. sende asynchrone Nachricht, warte auf ggf. Asynchron ankommende Antwort auf anderem Kanal)

Messaging über MicroProfile

- Abstraktion vom konkreten Messaging-System
- Nicht alles konfigurierbar
- Ggf. Messaging-spezifische Konfigurationen möglich oder sogar nötig
 - Z.B. `value.(de)serializer` für Kafka oder `address` zum Konfigurieren des `queue-names` für JMS
- Schreibende und lesende Methoden werden über Annotationen (`@Outgoing`, `@Incoming`) an das entsprechende Messsaging-System gekoppelt

Messaging über MicroProfile

- Schreibende und lesende Methoden werden über Annotationen (@Outgoing, @Incoming) an das entsprechende Messaging-System gekoppelt
 - Werden automatisch aufgerufen
 - @Outgoing einmalig
 - @Incoming für jede Nachricht
 - @Outgoing-Methoden dürfen keine Parameter haben

Messaging über MicroProfile

- Programmatisches Senden einer Nachricht kann über Emitter erfolgen
 - Emitter werden als Parameter mit `@Channel` injected und durch diese Annotation an das entsprechende System gekoppelt
 - Anwendungsfall: erhalte Entität per REST-Call, sende diese Entität auf eine Messaging-queue



Messaging in Quarkus

Messaging in Quarkus

- Erweiterungen sind messaging-system Spezifisch:
 - AMQP: [quarkus-smallrye-reactive-messaging-amqp](#)
 - Kafka: [quarkus-smallrye-reactive-messaging-kafka](#)
- Basiskonfigurationen für beide Erweiterungen gleich
- Jedes Messaging-System braucht ggf. noch Zusatzkonfiguration



Code!



Was noch?

Was noch?

- Auch opentracing/opentelemetry ist Teil von Microprofile → Tracing durch Messaging-Systeme hindurch möglich
 - Gerade im Umbau, viel Boilerplate nötig, wird sich voraussichtlich mit OpenTelemetry ändern
- Kafka-Streams
- Spezialisiertere Extensions, “an MicroProfile vorbei”, z.B. durch [quarkus-qpuid-jms](#)
- Integration mit Camel



Fragen?



Vielen Dank!



Dr. Marco Bungart
Senior Software Engineer

Büro Düsseldorf
Kanzlerstraße 8
D-40472 Düsseldorf
Tel.: +49-89-45841-100
Marco.Bungart@consol.de
www.consol.de
Twitter: @turing85