HIBERNATE REACTIVE & SPRING WEBFLUX

Andrew Rembrandt,
andrew@3ap.ch

3apedia '22
12 May, 2022

HOUSEKEEPING

- Today
  - Talk, then workshop part 1
- Tomorrow
  - Workshop part 2

ANOTHER REACTIVE TALK?

- Quarkus, Vert.x, & micronaut all have essential reactive features
- Spring (reactive) is *still* missing one or two things
- Hibernate has the kitchen sink, but;
  - Blocking is bad™
- Hibernate Reactive has many enterprise features (ADD, relationships, @Filter etc)

REACTIVE DATA

- R2DBC - development started end of 2017
  - Pivotal / VMWare Tanzu driven
- Hibernate Reactive - development started early 2020
  - Eclipse / Vert.x driven
  - Spawned out of:
    - Reactive PG Client (development started early 2017); and
    - 'Reactiverse' community

R2DBC

- Similar Spring JPA/Data Repository feel, but with Mono / Flux
    - Doesn't support relations!
        - Pivotal/Tanzu dropped the ball (3 years+): https://github.com/spring-projects/spring-data-r2dbc/issues/99#issuecomment-610783285
        - Even Josh Long is silent: https://twitter.com/gabac/status/1494635596832247828
            - Tanzu, speaker, & author of Reactive Spring

MUTINY!

- https://smallrye.io/smallrye-mutiny/
- A *simple* / clean api (compared to Project Reactor and RxJava)
  - Removes cognitive overload for building an asynchronous system
  - `Uni` (Monad for single item / error)
  - `Multi`
- Reactive choice in Quarkus (Vert.x can easily be adapted)

MUTINY & SPRING

- Two approaches:
  - Use mutiny helper methods to convert to Mono/Flux

```
<myUni>.convert().with(toMono())
```

  - Wholesale with a reactive adapter:

```
public class MutinyAdapter {
private final ReactiveAdapterRegistry registry;
@PostConstruct
public void registerAdapters(){
    registry.registerReactiveType(
        ReactiveTypeDescriptor.singleOptionalValue(Uni.class,
        ()-> Uni.createFrom().nothing()),
        uni ->((Uni<?>)uni).convert().toPublisher(),
        publisher ->  Uni.createFrom().publisher(publisher)
    );
    registry.registerReactiveType(
        ReactiveTypeDescriptor.multiValue(Multi.class, ()-> Multi.createFrom().empty
        multi -> (Multi<?>) multi,
        publisher-> Multi.createFrom().publisher(publisher));
}
}
```

MUTINY IS CLEAN

- Clear separation into factory classes:

```
Uni.createFrom().failure(new RuntimeException("Goodbye World!"))
```

- Reactor `Mono`/`Flux` is polluted and not in a consistent FP style

```
Mono.error(new RuntimeException("Hopefully there's no `onErrorResume()`..."))
```

  - Uni.java: 882 (straightforward) lines
  - Mono.java: 5206 (not really readable) lines

DOES HIBERNATE REACTIVE / MUTINY HAVE X?

- No derived query methods OOB
  - Spring Data's logic can be reused with some 'smarts':
    - github.../spring-data-commons/.../repository/query/parser
    - github.../spring-data-jpa/.../repository/query/JpaQueryCreator.java
- Mutiny has 'context'

```
Context context = Context.of("X-CUSTOMER-ID", customerId);

pipeline.withContext((multi, ctx) -> multi.onItem()
        .transformToUniAndMerge(item -> makeRequest(item, ctx.get("X-CUSTOMER-ID"))))
        .subscribe().with(context, item -> handleResponse(item), err -> handleFailure(e
```

https://smallrye.io/smallrye-mutiny/guides/context-passing

WORKSHOP 1A OVERVIEW

- Spring Webflux & Hibernate Reactive w/ testcontainers
    - Functional REST API CRUD Orders & Products app
    - Currently in Webflux & R2DBC
    - Convert it to Hibernate Reactive
        - Dependencies are already included

OPEN YOUR IDE...

- https://github.com/andrewrembrandt/hibernate-reactive-spring-talk