

Spring vs Quarkus: Framework comparison

Vinicius Antonio Gai

<https://www.linkedin.com/in/viniciusagai/>

<https://github.com/viniciustoni>





It's not YET covered

- Performance comparison
 - It's hard to have conclusions into simple apps
 - Probably I'll never do it in here, the diff is not that big and each case is one case
- Reactive Programming
 - It's a future plan for this presentation/code base
 - Honestly speaking: No time yet for it and Mutiny makes the code quite “dirty”
- Native images
 - Mostly because my Spring project was not fully yet prepare for that
 - KUDOS to Quarkus, because it was easy and almost no issues on run it with Native image

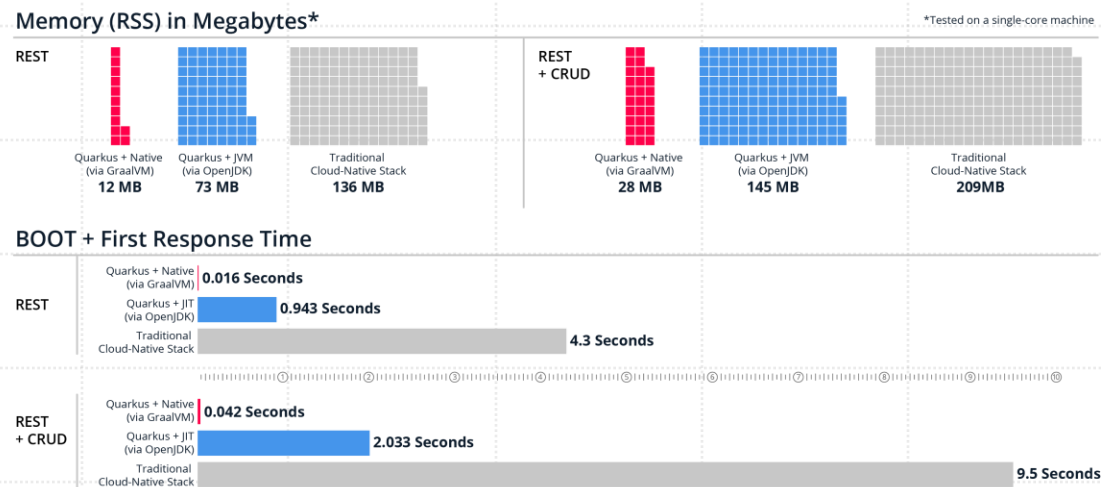


Agenda

- What is Quarkus
- Spring vs Quarkus- Overview
- From Spring to Quarkus: How did I approach it?
- Challenges/Problems
- Code demo
- What I liked on Quarkus

What is Quarkus

- A modern, kubernetes-native Java framework
- Optimized for fast startup times and small memory footprint
 - Native image support





Spring vs Quarkus – Overview

- Developer: VMWare (Pivotal)
- Initial release: June 2004
 - Spring boot at April 2014
 - Fully native image support: Nov 2022
- Active community with an extensive documentation and resources
- Spring modules
- ComponentScan/Proxy at runtime
 - Flexible at runtime, however decrease performance
- Developer: RedHat
- First release: March 2019
 - Already similar to “spring-boot”
 - Fully native image support from beginning
- Quickly increasing popularity
- Powerfull dev tools
- Quarkus extensions
- Build-time static analysis and proxy creation
 - Increases the startup time



From Spring to Quarkus: How did I approach it?

- Thinking on a Spring way
 - How to create beans?
 - How are the scope of the beans?
 - Write the code in spring and then convert to Quarkus
- Searching for already existing plugins/extensions/modules
- Using only Quarkus libs
 - Avoiding spring extensions: <https://quarkus.io/extensions/?search-regex=spring>
- Simple PoC app with RestAPI that connects to the database



Challenges/Problems

- Documentation is good. However, they are not cover complex scenarios and the community is still growing
- Not a big community in comparison with Spring
- Database:
 - Dynamic procedure calls
 - spring-jdbc is more mature than Quarkus one – It requires custom code to work the same way
 - Mongo
 - Spring-mongo is also more mature
 - Projections requires new DTOs/Entities



Code demo

- GitHub with code: https://github.com/viniciustoni/spring_vs_quarkus
- Search for `// DIFF:` on the code for some comments about the differences



What I liked on Quarkus

- Using Jakarta specification
- AdminUI – Without any extra dependency or configuration
- Quarkus dev:
 - It runs under maven/gradle/cli
 - Live reload with background compilation
 - Free of charge
- Quarkus extensions
 - From one side: I don't like so much due the coupling with the framework
 - BUT form the other side: The integration with other extensions is much smother plus its native image friendly



What I liked on Quarkus

- Hibernate Panache
 - Optimized for native images
- Integration with Testcontainers on tests
 - Requires us to clean up the jdbc.url, however only this change is needed
- How we override beans on test using @Mock instead of @Bean / @Primary on spring



Questions



References

- <https://quarkus.io/>
- <https://spring.io/>
- <https://quarkus.io/blog/quarkus-for-spring-developers/>
- <https://www.geeksforgeeks.org/difference-between-spring-boot-and-quarkus/>
- <https://quarkus.io/guides/hibernate-orm-panache>