Size does matter!

The battle of the Microframeworks

Java Forum Stuttgart 2019

Christian Schwörer
Novatec Consulting GmbH
JVM-Microframeworks on the rise
JVM-Microframeworks on the rise

MICROFRAMEWORKS...

MICROFRAMEWORKS EVERYWHERE!
What is this session about?

- (Subjective) **overview** on current JVM-Microframeworks
- Practical **introduction**
- Possible **fields of application** for microframeworks – and where not to use them (yet)
- **And what you won’t get:**
  - Deep dive in a single framework
  - Silver Bullet
Characteristics of Microframeworks (1/2)

- Designed for cloud-native microservice architectures
- Lightweight web framework
  - Start and configure a Server Engine (Tomcat, Jetty, Netty, etc.)
  - Provide REST endpoints
  - Deliver web content
- Resource efficiency
  - Fast startup time
  - Low memory consumption
Focus on simplicity and speed of development

Usually lack some advanced features (e.g. extended security, monitoring, multiple database abstractions)

GraalVM friendly
Example scenario

SPA / Browser \(\xrightarrow{\text{REST JSON}}\) Football-manager \(\xrightarrow{}\) MongoDB
Micronaut

"A modern, JVM-based, full-stack framework for building modular, easily testable microservice and serverless applications"

- Numerous modules (Kafka, SQL, NoSQL, Micrometer, AWS, etc.)
- Cloud-native modules for Service Discovery, Circuit Breakers, Distributed Tracing etc.
- Fast startup and low memory consumption
  - Uses Ahead of Time (AOT) compilation
  - Reflection free, runtime proxy free and no dynamic classloading
- Current version: 1.1.3
“Talk is cheap. Show me the code.”

Linus Torvalds
Javalin

“Javalin is a very lightweight web framework for Kotlin and Java [...] . Javalin’s main goals are simplicity, a great developer experience, and first class interoperability between Kotlin and Java.”

- Runs on top of Jetty
- Very simple: only few concepts that need to be learned
- Primarily blocking – as this is the easiest programming model (but may be switched into an asynchronous mode)
- Current version: 3.1.0
“Spring Fu is an incubator for Kofu (Ko for Kotlin, fu for functional), which provides a Kotlin API to configure Spring Boot applications programatically.”

- Explicit, functional configuration via Kotlin DSL instead of annotations
- Minimal set of features enabled by default
- Faster startup and lower memory consumption
  - No classpath scanning, Minimal reflection and annotation usage
  - Pure lambdas, no CGLIB proxy
- Current version: 0.1
## Recap

<table>
<thead>
<tr>
<th>Framework</th>
<th>Characteristics</th>
<th>Startup time*</th>
<th>Memory**</th>
<th>JAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micronaut</td>
<td>- Numerous modules</td>
<td>~ 1.1 s</td>
<td>~ 280 MB</td>
<td>~ 19 MB</td>
</tr>
<tr>
<td></td>
<td>- AoT compilation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Javalin</td>
<td>- Lightweight</td>
<td>~ 0.2 s</td>
<td>~ 210 MB</td>
<td>~ 11 MB</td>
</tr>
<tr>
<td></td>
<td>- Focus on simplicity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ktor</td>
<td>- Asynchronous</td>
<td>~ 0.6 s</td>
<td>~ 240 MB</td>
<td>~ 18 MB</td>
</tr>
<tr>
<td></td>
<td>- Kotlin coroutines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SpringFu</td>
<td>- Explicit configuration</td>
<td>~ 0.7s</td>
<td>~ 250 MB</td>
<td>~ 29 MB</td>
</tr>
<tr>
<td></td>
<td>- Spring Boot’s future?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Starting the example application as (Java 8-)JAR on a MacBook Pro

** Allocated heap size after start (not constrained with –Xmx or –Xms)
Possible fields of application

- Where microframeworks make sense:
  - (Dockerized) Microservices with dynamic load scenarios
  - Serverless Functions (but obviously not the webserver part)
  - API Gateways (non-blocking frameworks)
  - Mocking services

- And where they might not be appropriate (yet):
  - CloudFoundry as PaaS (because of its outstanding Spring Boot integration)
  - Standardization: choose one framework to rule them all
Further information

- @csh_0711
- https://github.com/csh0711/jvm-microframeworks-kotlin-samples
- https://blog.novatec-gmbh.de
- Frameworks
  - https://micronaut.io
  - https://javalin.io
  - https://ktor.io
  - https://github.com/spring-projects/spring-fu