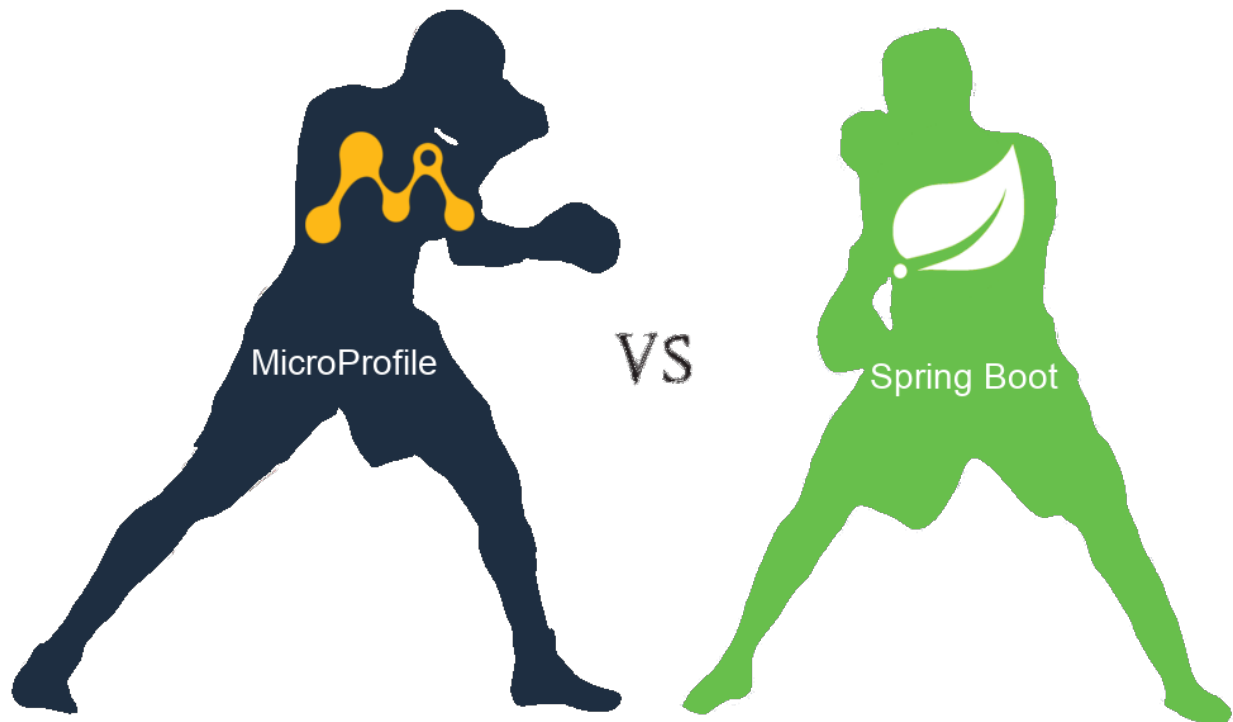


Java EE + MicroProfile - das bessere Spring Boot?



anderScore

 @anderScoreGmbH  anderScore.company  Java_Meetup_anderscore

Die Kontrahenten

- Der Champion: **Spring Boot**
 - Automagische Konfiguration
 - Starter für verschiedenste Anwendungsfälle
 - Embedded Tomcat, Jetty oder Undertow
 - Zugriff auf gesamtes Spring Ökosystem
- Der Herausforderer: **MicroProfile**
 - Robuste Standards und Implementierungen
 - Bulletproof
 - JEE/Jakarta Ökosystem

Die Kampfrichter



- **Daniel Krämer**
- Software-Entwickler, Architekt
- Integration und Migration
- Web Engineering
- Testautomatisierung
-  dkraemer-anderscore



- **Maik Wolf**
- Software-Entwickler
- Fullstack & Devops
- JEE/Jakarta - Fanboy
-  @da_mwolf
-  maikwolf

Der Boxstall

- Standort: Köln (mit Rheinblick...)
- Individuelle Softwareentwicklung
- Consulting und Festpreis
- Gesamter Application Life Cycle
- Konferenzen und Artikel
- Öffentliche Trainings



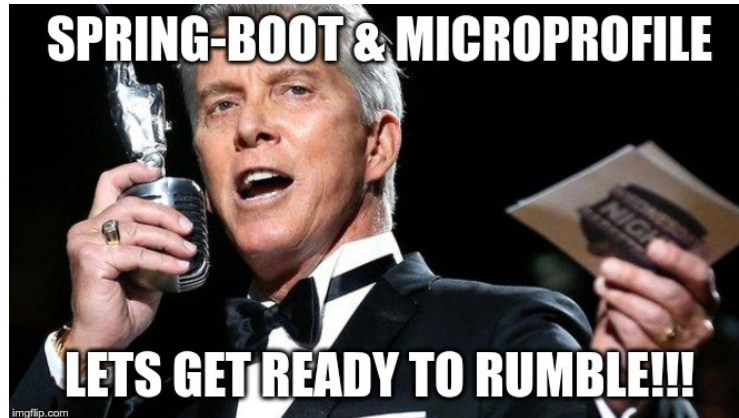
- Technologien
 - JEE, Spring
 - Wicket, Angular
 - Docker, Kubernetes, Apache Kafka
 - ...
- Goldschmiede@anderScore



Die Kriterien

1. Small runnable application
2. Externe Konfiguration
3. REST Endpoints
4. Health Check
5. Metriken

Auf in den Ring!



Runde 1 - Small runnable application

Small runnable application



Runde 1 - Small runnable application - Spring

Philosophie:

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We take an opinionated* view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration.

— <https://spring.io/projects/spring-boot>

*: eigensinnig, rechthaberisch

Runde 1 - Small runnable application - Spring

- Aufsetzen eines Projektes
 - Spring Initializr
 - CLI
 - IDE (Plugin)
- Projektstruktur:

```

Maven > spring [boot] [devtools] [microprofilevsspring master]
  src/main/java
    com.anderscore.spring
      Application.java
  src/main/resources
    application.properties
    
```

Runde 1 - Small runnable application - Spring

Starten der Anwendung:

```

@SpringBootApplication
public class Application {

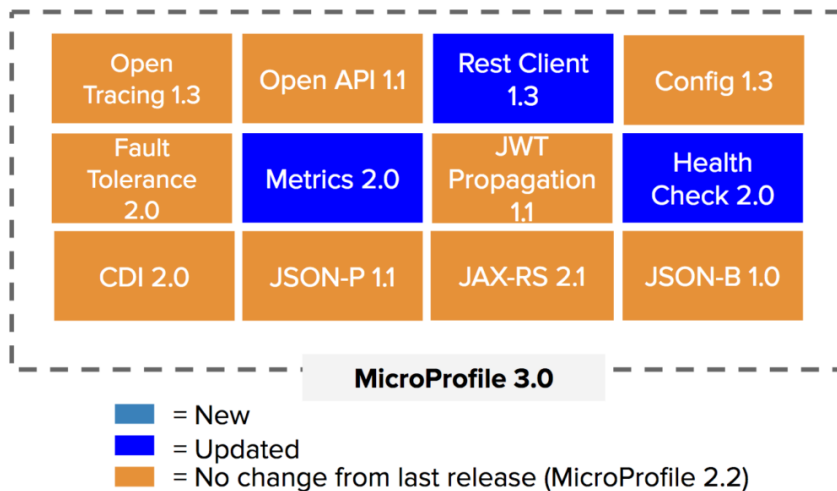
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
    
```

```
mvn spring-boot:run
```

Runde 1 - Small runnable application - MicroProfile



- Sammlung von Spezifikationen
- Fokus auf Microservice-Entwicklung



Runde 1 - Small runnable application - MicroProfile

Version 3.0



Version 2.2



Runde 1 - Small runnable application - MicroProfile



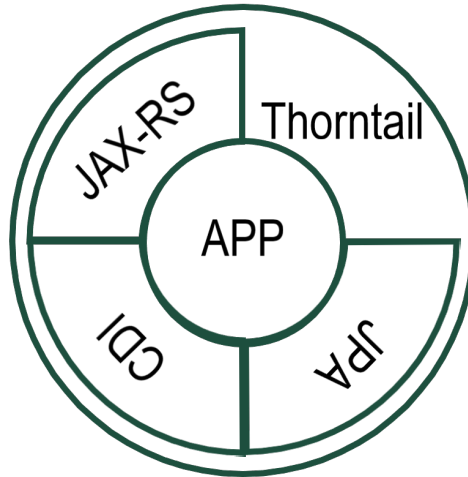
Thorntail offers an innovative approach to packaging and running Java EE applications by packaging them with just enough of the server runtime to "java -jar" your application. It's MicroProfile compatible, too. And, it's all much, much cooler than that ...

— <https://thorntail.io/>

Runde 1 - Small runnable application - MicroProfile

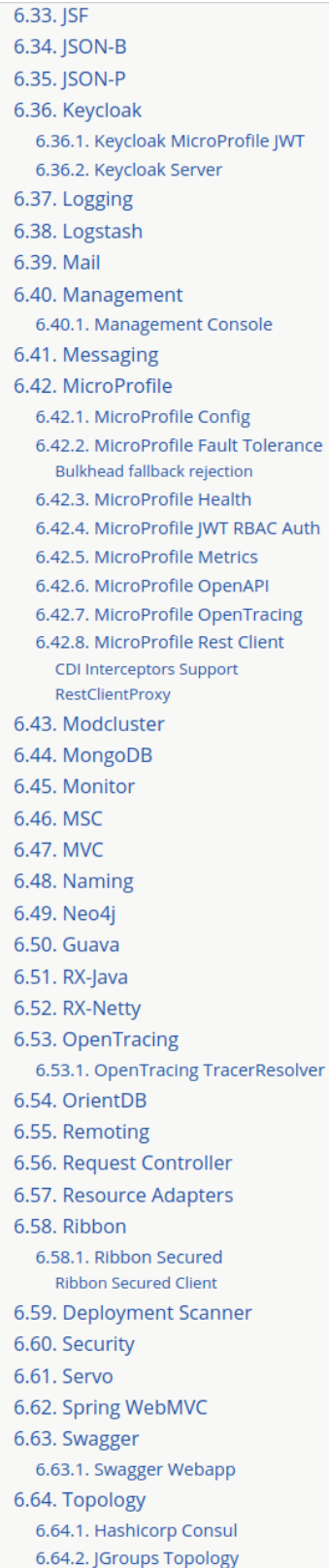
Thorntail

- "Just enough application server"
- Wildfly-Komponenten
- MicroProfile 3.0 (ab Version 2.5.0)
- Eine Menge Fractions



Runde 1 - Small runnable application - MicroProfile

70 Fractions (Stand: Version 2.5.0)



- 6.33. JSF
- 6.34. JSON-B
- 6.35. JSON-P
- 6.36. Keycloak
 - 6.36.1. Keycloak MicroProfile JWT
 - 6.36.2. Keycloak Server
- 6.37. Logging
- 6.38. Logstash
- 6.39. Mail
- 6.40. Management
 - 6.40.1. Management Console
- 6.41. Messaging
- 6.42. MicroProfile
 - 6.42.1. MicroProfile Config
 - 6.42.2. MicroProfile Fault Tolerance
 - Bulkhead fallback rejection
 - 6.42.3. MicroProfile Health
 - 6.42.4. MicroProfile JWT RBAC Auth
 - 6.42.5. MicroProfile Metrics
 - 6.42.6. MicroProfile OpenAPI
 - 6.42.7. MicroProfile OpenTracing
 - 6.42.8. MicroProfile Rest Client
 - CDI Interceptors Support
 - RestClientProxy
- 6.43. Modcluster
- 6.44. MongoDB
- 6.45. Monitor
- 6.46. MSC
- 6.47. MVC
- 6.48. Naming
- 6.49. Neo4j
- 6.50. Guava
- 6.51. RX-Java
- 6.52. RX-Netty
- 6.53. OpenTracing
 - 6.53.1. OpenTracing TracerResolver
- 6.54. OrientDB
- 6.55. Remoting
- 6.56. Request Controller
- 6.57. Resource Adapters
- 6.58. Ribbon
 - 6.58.1. Ribbon Secured
 - Ribbon Secured Client
- 6.59. Deployment Scanner
- 6.60. Security
- 6.61. Servo
- 6.62. Spring WebMVC
- 6.63. Swagger
 - 6.63.1. Swagger Webapp
- 6.64. Topology
 - 6.64.1. Hashicorp Consul
 - 6.64.2. JGroups Topology

Runde 1 - Small runnable application - MicroProfile


```
<project>
  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>io.thorntail</groupId>
        <artifactId>bom-all</artifactId>
        <version>${version.thorntail}</version>
        <scope>import</scope>
        <type>pom</type>
      </dependency>
    </dependencies>
  </dependencyManagement>

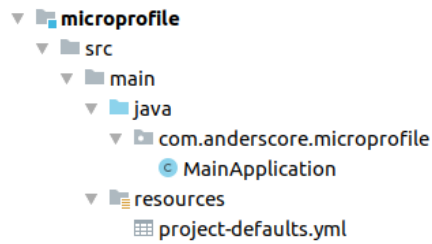
  <dependencies>
    <dependency>
      <groupId>io.thorntail</groupId>
      <artifactId>microprofile</artifactId>
    </dependency>
  </dependencies>
```

```
<build>
  <finalName>Microprofile</finalName>
  <plugins>
    <plugin>
      <groupId>io.thorntail</groupId>
      <artifactId>thorntail-maven-plugin</artifactId>
      <version>${version.thorntail}</version>

      <executions>
        <execution>
          <goals>
            <goal>package</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
</project>
```

Runde 1 - Small runnable application - MicroProfile

- **Aufsetzen eines Projektes**
 - MicroProfile Starter **oder** Thorntail Project Generator
 - CLI
- **Projektstruktur:**



Runde 1 - Small runnable application - MicroProfile

Starten der Anwendung:

```

@ApplicationPath("/")
public class MainApplication extends Application {

    public MainApplication() {
    }

}

```

```

$ mvn package && java -jar ./target/microprofile-thorntail.jar
$ mvn thorntail:run

```

Runde 1 - Small runnable application - MicroProfile

| | | | |
|---|------------------------------|----------|---------------------|
| Installed fraction: | MicroProfile | - STABLE | |
| io.thorntail:microprofile:2.5.0.Final | | | |
| Installed fraction: | JAX-RS | - STABLE | |
| io.thorntail:jaxrs:2.5.0.Final | | | |
| Installed fraction: | Logging | - STABLE | |
| io.thorntail:logging:2.5.0.Final | | | |
| Installed fraction: | Undertow | - STABLE | |
| io.thorntail:undertow:2.5.0.Final | | | |
| Installed fraction: | Elytron | - STABLE | |
| io.thorntail:elytron:2.5.0.Final | | | |
| Installed fraction: | CDI | - STABLE | |
| io.thorntail:cdi:2.5.0.Final | | | |
| Installed fraction: | CDI Configuration | - STABLE | io.thorntail:cdi- |
| config:2.5.0.Final | | | |
| Installed fraction: | Bean Validation | - STABLE | io.thorntail:bean- |
| validation:2.5.0.Final | | | |
| Installed fraction: | Transactions | - STABLE | |
| io.thorntail:transactions:2.5.0.Final | | | |
| Installed fraction: | JAX-RS with JSON-P | - STABLE | io.thorntail:jaxrs- |
| jsonp:2.5.0.Final | | | |
| Installed fraction: | JAX-RS with JAXB | - STABLE | io.thorntail:jaxrs- |
| jaxb:2.5.0.Final | | | |
| Installed fraction: | MicroProfile Config | - STABLE | |
| io.thorntail:microprofile-config:2.5.0.Final | | | |
| Installed fraction: | MicroProfile Health | - STABLE | |
| io.thorntail:microprofile-health:2.5.0.Final | | | |
| Installed fraction: | Management | - STABLE | |
| io.thorntail:management:2.5.0.Final | | | |
| Installed fraction: | MicroProfile Fault Tolerance | - STABLE | |
| io.thorntail:microprofile-fault-tolerance:2.5.0.Final | | | |
| Installed fraction: | Hystrix | - STABLE | |
| io.thorntail:hystrix:2.5.0.Final | | | |
| Installed fraction: | MicroProfile JWT RBAC Auth | - STABLE | |
| io.thorntail:microprofile-jwt:2.5.0.Final | | | |
| Installed fraction: | MicroProfile Metrics | - STABLE | |
| io.thorntail:microprofile-metrics:2.5.0.Final | | | |
| Installed fraction: | MicroProfile OpenAPI | - STABLE | |
| io.thorntail:microprofile-openapi:2.5.0.Final | | | |
| Installed fraction: | MicroProfile OpenTracing | - STABLE | |
| io.thorntail:microprofile-opentracing:2.5.0.Final | | | |
| Installed fraction: | MicroProfile Rest Client | - STABLE | |
| io.thorntail:microprofile-restclient:2.5.0.Final | | | |

Runde 2 - Externe Konfiguration

Externe Konfiguration



Runde 2 - Externe Konfiguration - Spring

Profiles + Properties:

```
@Configuration
@PropertySource("classpath:application-${spring.profiles.active:dev}.properties")
@Import({PersistenceConfig.class, SecurityConfig.class})
public class AppConfig {
}
```

▼ > src/main/resources

- application-dev.properties
- > application-prod.properties
- application-test.properties

```
mvn spring-boot:run -Dspring.profiles.active=test
```

Runde 2 - Externe Konfiguration - MicroProfile

▼ resources

- project-defaults.yml
- project-prod.yml
- project-test.yml

Shell

```
$ java -jar myapp-thorntail.jar -Stest
```