



Quarkus 3

Das Jakarta EE Runtime Framework

Markus Pauer, GEDOPLAN GmbH



Markus Pauer

- ≡ GEDOPLAN GmbH, Bielefeld
 - ≡ GEDOPLAN IT Consulting
Softwareentwicklung, Beratung, Konzepte, Reviews
 - ≡ GEDOPLAN IT Training
Java, JEE, Tools u.v.a.m. in Berlin, Bielefeld, on-site
- ≡ Trainer und Berater
- ≡ 20 Jahre Erfahrung im Jakarta EE Umfeld
- ≡ Full-Stack-Entwickler



Was ist Quarkus eigentlich?

- ≡ Jakarta EE Runtime Framework
- ≡ Fokus auf Container
- ≡ Basiert auf Jakarta EE Standards
- ≡ Erweitert durch Community Projekte
- ≡ Entwicklerfreundlich



Jakarta EE Standard

- Jakarta EE 10 wurde im September 2022 veröffentlicht
- Quarkus 2 basiert auf Jakarta EE 8
- Quarkus 3 basiert auf Jakarta EE 10
- Im November 2022 wurde die erste Quarkus 3 Alpha Version veröffentlicht
- Quarkus 3.0.0.Final wurde im April 2023 veröffentlicht
- Aktuell Quarkus 3.4.1.Final (21.09.2023)

Context and Dependency Injection

- Quarkus ist ein Build-Time-Framework
- Optimierungen bereits zur Compilezeit
- Bisher nicht vollständig kompatibel zu Jakarta CDI
 - Portable Extensions nicht unterstützt
- Eigenständige CDI-Implementierung (ArC)
 - Basiert in weiten Teilen auf Jakarta EE
 - Veränderte Bean-Discovery
 - Sinnvolle Erweiterungen des Standards

Jakarta EE 10

- CDI Spezifikation wurde aufgeteilt in Core, SE und EE
- Core enthält zwei Ansätze Lite und Full
- Application Server verwenden CDI Full
- CDI Lite für die Runtime Frameworks
 - Kein Session und Conversation Scope
 - Keine Dekoratoren
 - Keine Vererbung und Spezialisierung von Beans
 - Nur Bean Discovery Mode annotated
- Blogartikel zu CDI -> <https://quarkus.io/blog/on-the-road-to-cdi-compatibility/>

Quarkus CDI Erweiterungen

- Simplified Constructor Injection
- Removing unused Beans
- Default Beans
- Enabling Beans for Profiles or Properties
- Simplified Producer Method Declaration
- Injecting multiple Bean Instances
- ...

Neues Dev UI

- ≡ Vollständig überarbeitet
- ≡ Erweiterbarkeit verbessert

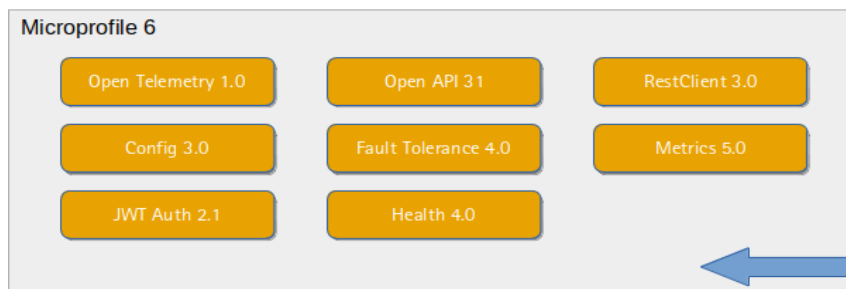
The screenshot displays the 'Dev UI' interface for Quarkus, specifically the 'Extensions' section. The interface is organized into a grid of extension cards, each providing details about a specific tool or library. A sidebar on the left offers navigation options like 'Configuration', 'Continuous Testing', 'Dev Services', and 'Build Metrics'. The top right corner indicates the current version as 'quarkus3-frontend 1.0-SNAPSHOT'.

Extension Name	Description	Version	Status
Apache POI	Read and write files in Microsoft Office formats, such as Word, Excel and PowerPoint	5.2.3	PREVIEW
ArC	Build time CDI dependency injection	69 (Beans), 2 (Observers), 70 (Removed components)	
MyFaces Core	MyFaces Core is a JSF / Faces implementation which also provides a Quarkus integration.	4.0.1	
PrimeFaces	PrimeFaces - is the ultimate component suite for JavaServer Faces	13.0.0	
iText	Create and manipulate PDFs on the fly	2.1.7	
Eclipse Vert.x	Write reactive applications with the Vert.x API		
Mutiny	Write reactive applications with the modern Reactive Programming library Mutiny		
REST Client Classic	Call REST services		
SmallRye Context Propagation	Propagate contexts between managed threads in reactive applications		
WebSockets	WebSocket communication channel support		
WebSockets Client	Client for WebSocket communication channel		
YAML Configuration	Use YAML to configure your Quarkus application		
Undertow Servlet	Support for servlets		
JSON-P	JSON Processing support		
AWT	Enable AWT and Java2D usage		PREVIEW
REST Client Classic Jackson	Jackson serialization support for the REST Client		
Jackson	Jackson Databind support		

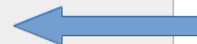
Quarkus 3.3.3

Microprofile 6

- ≡ Weiterentwicklung von Jakarta EE Standards
- ≡ Fokus auf Microservices



<https://microprofile.io>



Microprofile RestClient 3.0

- Quarkus Extension REST Client Reactive
- Implementiert durch RESTEasy Reactive

```
<dependency>  
  <groupId>io.quarkus</groupId>  
  <artifactId>quarkus-rest-client-reactive-jackson</artifactId>  
</dependency>
```

- API Definition über Interface
- Automatische Implementierung über `@RegisterRestClient`
- `@RestClient` injiziert die Client-Instanz

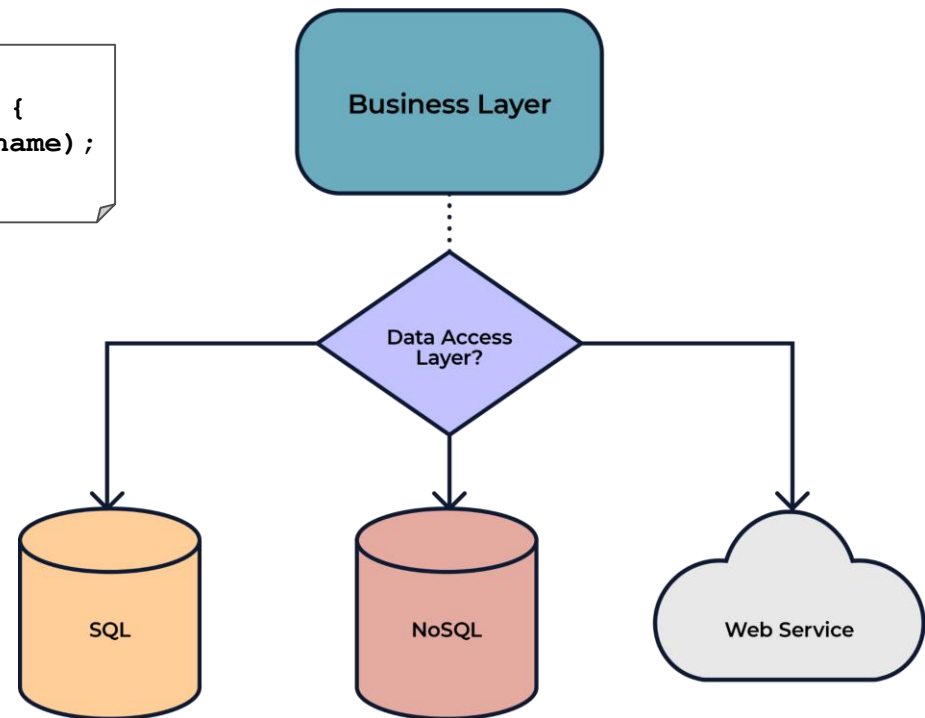
Jakarta Persistence mit Hibernate und Panache

- ≡ Vereinfachung der Persistenz
- ≡ Active Record oder Repository Ansatz
- ≡ Fertige CRUD-Methoden über Interface `PanacheRepository`
- ≡ Erweiterbar durch eigene Methoden
- ≡ Vereinfachte Definition von Abfragen

Jakarta Data Spezifikation

- <https://jakarta.ee/specifications/data/1.0/>
- Geplant für Jakarta EE 11

```
@Repository
public interface PersonRepository {
    List<Person> findByName(String name);
}
```



Quelle: https://jakarta.ee/specifications/data/1.0/data-1.0.0-b3#_repository

Migrationswerkzeug

- ≡ Umstellung auf Jakarta Namespaces
- ≡ Aktualisierung der Erweiterungen und Abhängigkeiten
- ≡ Verfügbar über Quarkus CLI, Maven und Gradle

```
$ quarkus update --stream=3.3
```

```
$ ./mvnw io.quarkus.platform:quarkus-maven-plugin:3.3.3.Final:update -N -Dstream=3.3
```

```
$ ./gradlew -PquarkusPluginVersion=3.3.3.Final quarkusUpdate --stream=3.3
```

Long-Term Release

- ≡ Normaler Release-Zyklus von ca. 5 Wochen wird beibehalten
- ≡ Aktuell drei unterstützte Versionen verfügbar
- ≡ Quarkus 2 → 2.16.11.Final
- ≡ Quarkus 3 → 3.4.1.Final
- ≡ Quarkus 3 LTS → 3.2.6.Final
- ≡ Erhält ein Jahr lang Aktualisierungen

Weitere Neuerungen

≡ Microprofile Open Telemetry

- ≡ Ersatz für Microprofile Open Tracing
- ≡ Guide → <https://quarkus.io/guides/opentelemetry>

≡ Hibernate ORM 6.2

- ≡ Jakarta Persistence nicht betroffen
- ≡ Migrationsanleitung → <https://github.com/quarkusio/quarkus/wiki/Migration-Guide-3.0:-Hibernate-ORM-5-to-6-migration>

- ≡ Blogartikel zu Quarkus 3 → <https://quarkus.io/blog/quarkus-3-0-final-released/>

Fazit

- ≡ Quarkus als die Alternative für Enterprise Server
- ≡ Aktualisierung auf Jakarta EE 10 und Microprofile 6
- ≡ Ideenschmiede für Jakarta EE Entwicklungen
- ≡ Leichtgewichtig und Entwicklerfreundlich
- ≡ Unterstützung bei der Migration auf Quarkus 3
- ≡ Beispiele → <https://github.com/GEDOPLAN/quarkus3>

Fragen ?