



Ktor

Kotlin HTTP Client und Server Framework

Was ist Ktor?



Kotlin basiertes HTTP Server (und Client) Framework

Asynchrone pipeline basierend auf Coroutines

OpenSource, maintained by JetBrains (<https://github.com/ktorio/ktor>)

Unterstützte targets:

- Standalone (JVM, Linux, macOS)
- Servlet container (Tomcat & Jetty)
- GraalVM



Ktor vs



- Client & Server
 - Kotlin only (+ Multiplattform support)
 - “Unabhängig” - sehr flexibel
 - Coroutine based
 - Keine Annotation magic
- Nur Server
 - eigentlich auf Java Basis - später Kotlin Support
 - Spring Ökosystem/Überladene Frameworks
 - Thread based
 - Annotation magic

MongoDB



NoSQL Datenbank

Basiert auf nicht relationalem Dokumentmodell (BSON/JSON)

Keine Beziehungen/Abhängigkeit von Daten (keine Fremdschlüssel)

Ermöglicht Replikation und Hochverfügbarkeitszenarien (Horizontaleskalierung)

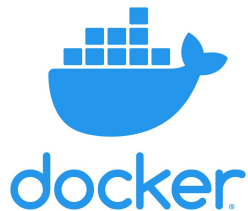
Keine Transaktionen (ACID)

Setup/Plugins



- Ktor
 - Core
 - Content Negotiation (JSON)
 - Auth + BCrypt
 - Default Headers
 - Logging
 - Status Page
 - Netty (Engine)
 - Tests
- MongoDB
 - Driver
 - Katerbase (“typesafe” kotlin wrapper für den Java-Driver)

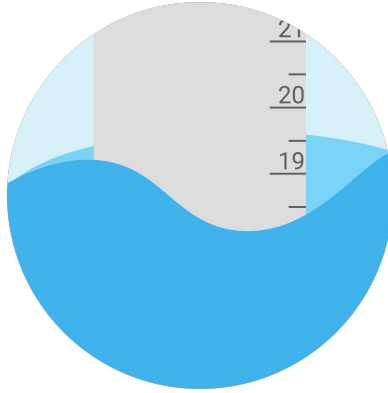
Testszenario & Probleme



- Gitlab Runner läuft mittels Docker Container
 - Jeder Job wird in einem eigenen Container ausgeführt

```
e2e-test-job:  
  image: openjdk:17-jdk-slim
```

- DB kann somit nicht in einem Container gestartet werden
 - Dind (Docker-in-Docker) ist nur für die Imageerstellung vorgesehen
- Provisorische Lösung: MongoDB Cloud Storage
- Besser:
 - DB mocken
 - In-Memory DB
 - Testcontainer → Benötigt aber einen GitLab Shell Runner



Water Level Monitor



Live demo