

Spring Boot

Ziel

- In wenigen schritten eine Applikation/(Micro)Service erstellen
- stand-alone ohne JEE-Container / Server
- production-ready
- ohne Code-Generierung
- nur mit Java
- mit aufeinander abgestimmten Libs / Abhängigkeiten

Was wird geschehen?

- Applikation / Service erstellt mit
 - Persistence
 - HTML und JSON Schnittstelle
 - Einfach stand-alone testbar
 - Security
- Management-Console / Shell

Was richtig geil ist...

.... Spring Boot ist
Microservice-Ready, durch die
Integration von Spring Cloud

Spring Cloud

- Sammlung von Lösungen für die Entwicklung und Betrieb von Microservices
 - Resilience (Ausfallsicherheit)
 - Service Registry & Service Recovery
 - Load balancing
 - Kommunikation
 - Routing
 - Konfiguration

Resilience Hystrix

- Spring Cloud Netflix
- Circuit Breaker - Implementierung
- Integration
 - Server
 - Client

Service Registry & Recovery

Eureka

- Spring Cloud Netflix
- Services registrieren und finden
- Integration
 - Server
 - Client

Load balancing Ribbon

- Spring Cloud Netflix
- Auffinden von Services und verteilen der Anfragen auf alle Instanzen
- Nutzt Hystrix, Eureka
- Integration
 - Client

Kommunikation Feign

- Spring Cloud Netflix
- REST-Kommunikation deklarativ
 - Über Annotation von Interfaces
- Nutzt Ribbon und Eureka
- Integration
 - Client

Routing Zuul

- Spring Cloud Netflix
- Reverse Proxy
- Integration
 - Server

Konfiguration

Archaius / Spring Cloud Config

- Archaius in Spring Cloud Netflix
- Zentrales Konfigurationsmanagement
- Automatische Anpassung der Konfiguration nach Änderung
- Integration
 - Client
 - Server