

## Agenda "Spring Boot / Docker für Sokabau"

Dr. Rainer Sawitzki, 2.2.2023

Teilnehmer: Anwendungsentwickler

Vorkenntnisse: Gute Kenntnisse von Java 11, JEE, Umgang mit der Entwicklungsumgebung IntelliJ

Methode: Vortrag, Präsentation, Diskussion, eigene Übungen

Dauer: 20 Unterrichtseinheiten mit jeweils 90 Minuten, verteilt auf 5 Tage. Davon 12 Einheiten Spring, 8 Einheiten Docker

Ort: Online, Videoplattform Teams

Sprache: Deutsch

Termin: Wird separat vereinbart

Ressourcen: PDF-Handout, Git-Repository mit Beispielen und Musterlösungen, digitales Flipchart

Hinweise: Die Präsentation erfolgt mit IntelliJ Community Edition

Technik: Ein lesender Zugriff auf die GitHub-Webseite <https://github.com/Javacream> ist möglich

Optional: Ein lesender Zugriff auf ein digitales Flipchart unter

[https://docs.google.com/document/d/1yjfrg30aAuIB\\_M9Z5E5uGPfIWq3ovgQ41K4AB0hE6gY/edit](https://docs.google.com/document/d/1yjfrg30aAuIB_M9Z5E5uGPfIWq3ovgQ41K4AB0hE6gY/edit) ist möglich

## **Inhalte Spring / Spring Boot:**

1. Einführung
  - a. Was ist Spring / Spring Boot? Abgrenzung und Gemeinsamkeiten zu JEE
  - b. Eine Plain Old Java Anwendung. Hinweis: Diese wird auf GitHub bereitgestellt und dient als Ausgangsbasis für die weiteren Blöcke
2. Spring Boot
  - a. Umstellen des Projekts auf Spring Boot
  - b. Funktions-Übersicht: Build-Prozess, Konfiguration, Autoconfig
3. Spring Core Teil 1: CDI
  - a. Das Design-Pattern Context & Dependency Injection
  - b. Spring-Annotations und JavaConfig
  - c. Staging, Profiles
  - d. Testing
4. Spring Core Teil 2: Aspektorientierte Programmierung
  - a. Grundlagen der AOP
  - b. Umsetzung mit AspectJ
  - c. Praxisbeispiele
5. Erweiterung der Anwendung
  - a. RESTful WebServices mit Spring MVC
  - b. Zugriff auf eine relationale Datenbank mit Spring Data JPA

## **Inhalte Docker**

6. Einführung
  - a. Virtualisierung, Container, Cloud
  - b. Docker-Architektur
7. Docker Container
  - a. Docker als Package Manager, der Docker Hub
  - b. Konfiguration eigener Container zur Bereitstellung von Servern und Infrastruktur
  - c. Docker Container und deren Konfiguration im Detail
  - d. Arbeiten mit Docker-Containern: docker cp, docker exec, Bindings und Volumes
8. Docker Images
  - a. Images vs Container
  - b. Anpassung von Images: commit und Dockerfile
  - c. Dockerfile-Syntax
  - d. Bauen von eigenen Images mit Schwerpunkt Java / Spring Boot, Aufbau einer effizienten Dockerfile-Hierarchie
  - e. Ausbringen der Images in ein Artefakt-Repository (Nexus / Artifactory)