Agenda für Development & Design of High Performing APIs

Dr. Rainer Sawitzki, 26.6.2020

Terminvorschlag: Noch offen

Ort: bevorzugt Online, Präsenztraining nach Absprache möglich

Zielgruppe: Entwickler

Vorkenntnisse: Erfahrene Spring-Boot Entwickler, Docker-Kenntnisse, Projekt-Erfahrung im Design von REST-Schnittstellen.

Methode:

- Vortrag, Präsentation, Diskussion, Übungen
- Hinweis zu den Übungen: Hier sind keine klassischen Programmier-Übungen etc. gemeint. Es werden typische Referenzanwendungen vorgestellt, analysiert und modifiziert.

Technische Voraussetzungen:

- Jeder Teilnehmer hat eine eigene Umgebung mit
  - Java-Installation (>= Java 8)
  - Entwicklungsumgebung Eclipse in einer einigermaßen aktuellen Version
  - Internet-Zugang
  - Freier Download von Java-Bibliotheken entweder über das Internet oder über ein internes Artefakt-Repository. Hier ist zu garantieren, dass die Entwickler korrekte Einstellungen (Proxy, Credentials) getroffen haben.
  - Docker-Installation entweder unter Windows 10 oder Linux. Auch hier muss der Download von Images direkt von Dockerhub oder einem internen Repository-Server gewährleistet sein.
- Digitale Durchführung bevorzugt mit einer Plattform des Kunden (Teams, Zoom, Skype) oder von Integrata-Cegos (Jitsi). Ein virtueller Klassenraum wird nicht benötigt oder bereitgestellt.
- Der Referent benötigt bei einer Präsenzdurchführung
  - Seminarraumausstattung mit Flipchart + Papier, 4farbigem Stiftsatz (möglichst neu) und Projektor mit HDMI-Anschluss.
  - Internet-Zugang über WLAN oder LAN. Ein Zugriff auf das interne Netzwerk ist nicht notwendig.

Dauer:

- 2 Tage mit jeweils 4 Unterrichtseinheiten mit jeweils 90 Minuten, 6 Stunden netto Seminarzeit pro Tag.
- Vorschlag für die Seminarzeiten: 9:00 - 16:15 mit zwei Kaffeepausen 15’ sowie Mittagspause 45’ 12:15-13:00.

Ausgangssituation:

- Der Kunde betreibt eine Microservice-Architektur mit Spring Boot-Anwendungen in einem Kubernetes/OpenShift-Cluster
- Das Deployment erfolgt über eine fertig implementierte CI/CD-Pipeline
- Sowohl Startverhalten als auch der Durchsatz der Anwendung wird vom Kunden als Verbesserungswürdig angesehen.

Inhalte

- Spring Boot als Plattform für Microservices: the Good, the bad, the ugly
- Werkzeuge und Verfahren zum frühzeitigen Erkennen von potenziellen Performance- und Durchsatz-Problemen, Integration in die Build-Pipeline
- Monitoring und Orchestrierung von Container-basierten Spring Boot-Anwendungen zum optimierten Betrieb in einer Cloud-Umgebung
- Best Practices für Microservices-Architekturen: Schneiden der Services, Inter-Service-Kommunikation (synchron versus asynchron), Datenkonsistenz
- Diskussion und Workshop auf Grundlage der vom Kunden realisierten Anwendung.