

Requirements Full Stack development JAVA.

Volgende onderdelen dienen gekend te zijn voor je mondelinge verdediging!

1. Unit Test alle logica in je services met Mockito, mock je repos.

- Falende testen zijn niet ok.
- @Ignored testen zijn niet ok.
- Testen met pointless assertions zijn niet ok, bvb
 - `assertThat(true).isEqualTo(true);`

2. Test je services met MockMVC en Testcontainers.org.

3. Werk je Use Cases uit volgens de opdracht.

4. Deployment

- a. Lokaal op je laptop, alles dockerized is geen vereiste.

5. Kennis vragen

- Concepten DI en IOC grondig kennen, kunnen uitleggen en toepassen Annotations kunnen uitleggen en toepassen
- Weten wat het Proxy pattern is en hoe het gebruikt wordt.
- Wat zijn de standaard features van Spring Cloud?
- Wat is de functie van deze componenten?
- Wat is het verschil tussen micro services (Spring Cloud) en Spring Boot?
- Hoe zou je communiceren tussen je componenten?
- Hoe wordt het proxy pattern toegepast in jouw architectuur?
- Spring Boot, hoe werken annotaties?
- Waarom pas je messaging toe?
- Welke verschillende manieren van messaging kan je toepassen of heb je gebruik in je implementatie?
- Op welke problemen stoot als je log4j zou gebruiken zoals in Spring Boot?
- Hoe heb je logging toegepast?
- Heb je je afgevraagd heb ik hier met 'duplication of data' te maken? En zo ja waar/wanneer?
- Hoe verwerkt de Gateway requests?
- Hoe heb je CORS problemen opgelost?
- Heb je load balancing toegepast? Wat is het?
- Wat bedoelen we met Service registration en Discovery?
- Verklaar Open-Feign?
- Wat zijn de voordelen van Open-Feign?
- Wat zijn de voordelen van Spring Cloud?
- Noem de service implementatie die zorgt voor Registration en Discovery?
- Wat is het verschil tussen Spring Cloud en Spring Boot?
- Wat zijn de meest gebruikte Spring Cloud annotaties?
- Hoe ga je om met modules in je project setup?
- Wat is Hystrix.. WTF, wat is dat?
- Wat is de Pivotal Cloud Foundry (PCF)?

- Welke implementatie heb je gebruikt voor Messaging?
- Heb je bepaalde end-point beveiligd en hoe?
- Hoe heb je je microservices getest?
- Hoe identificeer je je brouwer? 1 of andere vorm van security toegepast?

Security op Microservices is een uitgebreid onderwerp waarbij Token based security geen onderdeel is van het Java onderdeel. Doch kunnen we zelf een Basic HTTP authentication header toevoegen om Security af te dwingen waar nodig. Je kan ook je eigen header voorzien indien gewenst. Laat dit je niet beperken.

Samengevat: Je dient in staat te zijn de geziene onderdelen in de lessen (zie slides) volledig toe te passen (kunnen) en kennen!