



Service Discovery & Co

Übung 8

Ziel der Übung

- Configuration verteilen
- Services finden
- Leader-Election
- Out-Of-Scope: Health-Check

Consul

- Consul is a scalable and highly available tool developed by HashiCorp that helps distributed applications in many ways. It helps services to **discover** each other, performs **health checks** on them, acts as DNS service and even provides a reliable **key-value storage**.

Consul starten

- Agent-Mode: The Consul agent is the core process of Consul. The agent maintains membership information, registers services, runs checks, responds to queries, and more. The agent must run on every node that is part of a Consul cluster.
- Siehe Docker

Consul-UI

- <http://127.0.0.1:8500/ui>
- Für uns wichtig:
 - Services
 - Key/Value

Aufgabe 1: KeyValue

- Consul Projekt
- Persistieren des Config-Objects „Config“:
 - Int: MaximalConnectionsPerSecond
 - Uri: RemoteStockService
 - List<Guid>: Tokens
- Werte können beliebig sein → Serialisierung in YAML
- Key: mainConfiguration
- Nuget-Pakete (bereits im Projekt enthalten)
 - <https://www.nuget.org/packages/Consul>
 - <https://www.nuget.org/packages/YamlDotNet/>

Aufgabe 2: Leader

- Mehrere Services mit gleicher Aufgabe – wer ist der Master?
- <https://learn.hashicorp.com/consul/developer-configuration/elections>
- Starten von 10 Parallelen Tasks
- Ausgabe einer eindeutigen Id in jedem Task
- Nutzen von `AcquireLock` mit 3 Sekunden Lock
- Ausgabe von Id, wenn Lock erhalten wurde

Aufgabe 3: Service Discovery / Catalog

- Registrieren von **externen** Services
- Nutzen von `client.Catalog.Register`
- Registrieren von 3 Services:
 - `BillingService`
 - `OrderService`
 - `MailService`
- Abfrage der 3 Services und Ausgabe in Console
- Nutzen von – Ihrer Meinung nach – sinnvollen Parametern von `CatalogRegistration` (mindestens 2)

Fragen.txt

- Siehe Fragen in Fragen.txt