

ToiletPaper #105

Leichtgewichtige REST-Clients mit Feign (Feign is fine)

Autor: Magnus S. / Software Architect / Business Division Automotive Bavaria

✗ Problem

In der Microservices/REST-Welt wird für jede direkte Kommunikation zwischen Services ein eigener Client benötigt. Das Schreiben dieser Clients ist zwar einfach, aber wenig spannend, und führt außerdem zu Boilerplate-Code.

✓ Lösung

Der deklarative REST-client **Feign** (<https://github.com/OpenFeign/feign>) erlaubt es in Spring-Applikationen, diese Aufgabe mit minimalem Codeeinsatz zu erledigen.

Vorteile / Features:

- frei konfigurierbare Encoder/Decoder (JSON, XML, ...), Logger, ErrorHandler und Interceptors (z.B. für OAuth)
- kompatibel mit JAX-RS Annotationen, Spring Cloud, Hystrix, Spring MVC Annotationen, Spring Web `HttpMessageConverter`
- unterstützt Load Balancing (mittels Ribbon und Eureka)
- komplexere Clients können mit der Feign Builder API realisiert werden

➔ Beispiel

```
@FeignClient(name = "coffeeMachineClient", configuration = Config.class, url = "${myUrl}")
public interface ICoffeeMachineClient {

    @RequestMapping(method = POST, path = "/coffees")
    CupOfCoffeeDTO createCoffee(@RequestHeader("userId") Id uId, CupOfCoffeeDTO request);

    @RequestMapping(method = GET, path = "/coffees")
    List<CupOfCoffeeDTO> getCoffees();

    @RequestMapping(method = GET, path = "/coffees/{id}")
    CupOfCoffeeDTO getCoffeeById(@RequestParam("id") String coffeeId);

    @RequestMapping(method = PUT, path = "/coffees/{id}")
    CupOfCoffeeDTO updateCoffee(@RequestParam("id") String id, CupOfCoffeeDTO request);

    @RequestMapping(method = DELETE, path = "/coffees/{id}")
    void deleteCoffee(@RequestParam("id") String id); //who would do this?
}
```

→ weniger als 15 Zeilen Code für einen kompletten CRUD-Client

Der Einsatz von Feign lohnt sich vor allem, wenn viele unterschiedliche Requests zu einem bestimmten Server notwendig sind.

Achtung: Feign-Clients können nur mit Text-basierten Schnittstellen umgehen, nicht mit Binärdaten (z.B. Datei-Up/-Download)

+ Weiterführende Aspekte

- https://cloud.spring.io/spring-cloud-netflix/multi/multi_spring-cloud-feign.html