

ToiletPaper #105

Lightweight REST Clients with Feign (Feign is fine)

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

✗ Problem

Within the Microservices/REST “universe”, a separate client is required for each direct communication between services. Writing these clients is simple, but not very exciting, and leads to boilerplate code.

✓ Solution

The declarative REST-client **Feign** (<https://github.com/OpenFeign/feign>) allows in Spring applications to perform this task with minimal code usage.

Advantages / Features:

- freely configurable encoders/decoders (JSON, XML, ...), loggers, error handlers and interceptors (e.g. for OAuth)
- compatible with JAX-RS annotations, Spring Cloud, Hystrix, Spring MVC annotations, Spring Web `HttpMessageConverter`
- allows load balancing (with Ribbon and Eureka)
- more complex clients can be realized with the Feign Builder API

➔ Example

```
@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?
}
```

→ less than 15 lines of code for a complete CRUD client

Using Feign particularly pays off in case of many different requests having to be issued against a particular server.

Caution: Feign clients can't handle binary data (e.g. file upload/download), but only text-based interfaces!

+ Further Aspects

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