SYNCHRONOUS WITH FEIGNCLIENT IN SPRING BOOT

FOGUE KAMGA DILANE – SOFTWARE ENGINEER

AGENDA

- ❑ Concept Study
- ❑ Feign Client
- ❑ Feign Client Implementation
- ❑ Conclusion

Concept Study

In micro-service based applications, the services are considered as processes and are communicated using inter process communication protocols (IPC) such as HTTP, AMQP etc. There are many communication types based on the goal and scenario of communication. One of the most common type of communication used is single-receiver communication with a synchronous protocol like HTTP/HTTPS when invoking a regular Web API HTTP service.

For implementing a synchronous communication between microservices. We can use either Feignclient or restTemplate. But restTemplate has some disadvantages such as Too many lines of code for each connection and Duplication of code with no business logic.

In this presentation, we will cover synchronous IPC with Feignclient in Spring boot by implementing an CRUD for Users that get data from a public API.

Feign Client

Feign client is a wrapper library for http communications between micro services. Feign makes the developer life easier using declarative annotations avoiding to write the duplicate lines of code. For Spring boot applications, one can find open feign library to implement this declarative REST API Clients.

All you have to do is to define an interface with the method declarations and @FeignClient annotation.

Since, the Feign Client dependency is not available on Spring initializr , we must add it manually into our pom.xml file by retrieved it by follow this link of the maven central repository https://mvnrepository.com/artifact/io.github.openfeign/feign-core/11.10

Feign Client Implementation

❑ Introduction to the Project

We have decided to implement an user registration CRUD with just the Read functionalities. We will proceed as follow

- **Create an interface UserClient with the @FeignClient annotation as well as the url of our public API as paramater and a get method which return all users.**

Now we can generate dtos by copying and pasting our data from the public API to https://www.jsonschema2pojo.org/. This will generate 4 dtos classes as follow

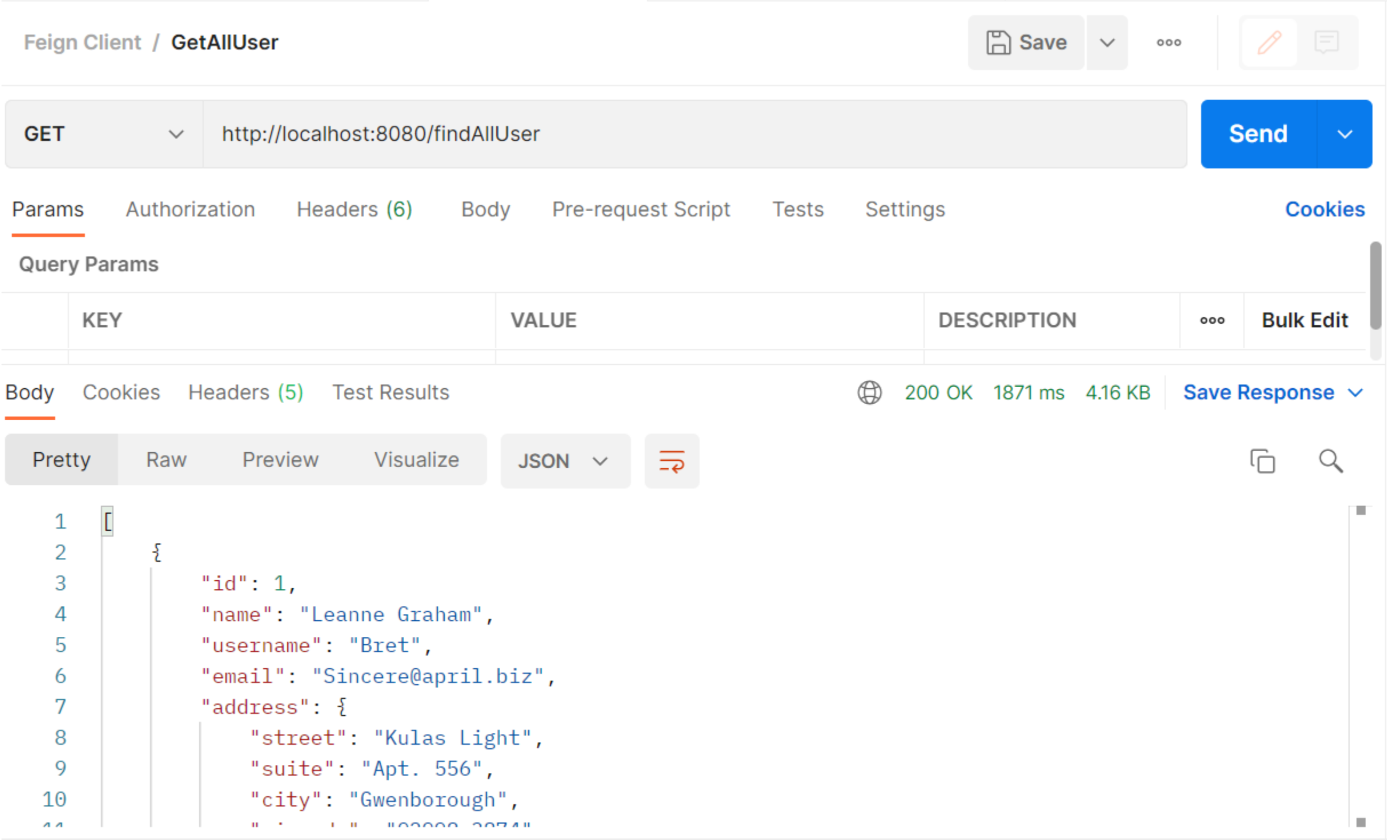
- **Generate a dto Adresse**
- **Generate a dto Company**
- **Generate a dto Geo**
- **Generate a dto UserResponse**
- **Add @RestController and @EnableFeignClients into our main class as well as findAllUser controller**

Now, let’s test our application using Postman

Feign Client Implementation

❑ Test our Feign Client Implementation

We will get all user, so our Feign Client implementation works



GET http://localhost:8080/findAllUser

200 OK 1871 ms 4.16 KB

```
[
    {
        "id": 1,
        "name": "Leanne Graham",
        "username": "Bret",
        "email": "Sincere@april.biz",
        "address": {
            "street": "Kulas Light",
            "suite": "Apt. 556",
            "city": "Gwenborough",
```

Conclusion

We have learned how to implement synchronous IPC with Feign Client into Spring Boot. The links of our resources and GitHub repository are attached below

https://www.youtube.com/watch?v=_MMf2SvNqxo

https://www.tutorialspoint.com/spring_cloud/spring_cloud_synchronous_communication_with_feign.htm#

https://medium.com/@nschsravanthi/micro-services-communication-with-feign-client-5ea3e60dbdd8

https://github.com/Dilane-Kamga/Synchronous-with-Feignclient.git

MERCI

Pour votre attention