**FEIGN**

* Các service có thể gửi các yêu cầu HTTP đến các endpoint của các service khác để thực hiện các tác vụ cụ thể. Trong Spring, để triển khai việc giao tiếp giữa các service qua HTTP, chúng ta có thể sử dụng các công cụ và thư viện hỗ trợ như RestTemplate, WebClient (từ Spring WebFlux) hoặc Feign
* Feign Client: Feign là một công cụ trong Spring Cloud giúp giảm bớt công sức trong việc giao tiếp giữa các service qua HTTP. Thay vì gọi các API bằng cách sử dụng RestTemplate truyền thống, chúng ta có thể tạo ra các interface với các phương thức tương ứng với các API của các service khác và Feign sẽ tự động tạo các yêu cầu HTTP cho chúng ta
* @FeignClient trong Spring:
  + Được sử dụng để đánh dấu một interface là một Feign Client. Điều này cho phép Feign tự động tạo các yêu cầu HTTP cho các API của các service khác
  + Trong trường hợp sử dụng server discovery (như Eureka), thuộc tính *name* sẽ được sử dụng để tìm service mục tiêu trong Eureka Server. Nếu không sử dụng server discovery, chúng ta có thể chỉ định URL cụ thể của service bằng cách sử dụng thuộc tính *url*

**Feign khi không sử dụng Service Discovery**

A screen shot of a computer

Description automatically generated

A computer code with black text

Description automatically generated

Tại service Client:

A white background with black text

Description automatically generated

A computer screen shot of a program

Description automatically generated

A computer code with black text

Description automatically generated

Tại service Example (localhost:8080):

A computer code with black text

Description automatically generated

**Feign với Service Discovery**

* Không cần sử dụng đến thuộc tính *url* nữa:

A computer code with black text

Description automatically generated

* Khi tích hợp Feign với một server Service Discovery (như Eureka, Consul, ZooKeeper hoặc Kubernetes), Feign sẽ tự động sử dụng thông tin từ service discovery để xác định danh sách các instance của dịch vụ mục tiêu. Sau đó, Feign sẽ sử dụng thư viện cân bằng tải mặc định của Spring Cloud (như Ribbon) để chọn một instance cụ thể của dịch vụ để gửi yêu cầu
* Cơ chế cân bằng tải mặc định trong Spring Cloud (Ribbon) thường sử dụng các thuật toán cân bằng tải như Round-Robin, Least Connection, hoặc Random để lựa chọn instance phù hợp. Việc này giúp tự động phân chia công việc giữa các instance của dịch vụ, làm giảm tải trên từng instance và tăng tính ổn định và mở rộng cho ứng dụng
* Điểm quan trọng là khi chúng ta sử dụng Feign, chúng ta không cần phải lo lắng về việc thực hiện cơ chế cân bằng tải thủ công. Spring Cloud sẽ tự động quản lý việc này, giúp chúng ta tập trung vào việc triển khai các phương thức gọi dịch vụ một cách dễ dàng và linh hoạt trong môi trường microservices