**Spring Cloud Routing-OpenFeign**

**🔍 ¿Qué es OpenFeign?**

**OpenFeign es una librería de cliente HTTP declarativo que te permite hacer llamadas REST a otros microservicios de manera súper sencilla y sin escribir mucho código.**

**Es como si convirtieras una interfaz Java en un cliente REST automático.  
¡Y si usas Eureka, además descubre el servicio por ti!**

**🧠 ¿Cómo funciona?**

**En vez de escribir esto 👇 (con RestTemplate o WebClient):**

Texto

El contenido generado por IA puede ser incorrecto.

**Con OpenFeign, simplemente haces:**

Texto

El contenido generado por IA puede ser incorrecto.

**Y luego:**

**🔗 ¿Qué integra Feign?**

* **✅ Descubrimiento de servicios con Eureka**
* **✅ Conversión automática de objetos JSON**
* **✅ Manejo de errores**
* **✅ Soporte para balanceo de carga (con Spring Cloud LoadBalancer)**
* **✅ Fallbacks (si usas Resilience4j o Hystrix)**

**AHORA SI INICIEMOS:**

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Para crear un openFeign usaremos los microservicios anteriores, que son eureka-server, eureka-cliente, pero ahora en este caso crearemos algo que consuma el microservicio de eureka cliente, para eso se usa openfeing, va consumir como si fuera un rest

Creamos el siguiente paquete:  
Texto

El contenido generado por IA puede ser incorrecto.

Primero en la clase principal escribimos estos:  
Texto

El contenido generado por IA puede ser incorrecto.

Luego en application.yml escribimos esto:  
Texto

El contenido generado por IA puede ser incorrecto.

Y luego en el consumidor feign es donde llamaremos al application.yml de eureka cliente:  
Texto

El contenido generado por IA puede ser incorrecto.

Sino te acuerdas el application.yml de eureka cliente es esto(no tocamos nada):  
Texto

El contenido generado por IA puede ser incorrecto.

Listo, eso es lo que hace el feign. Llama al microservicio cliente para consumirlo, luego pasamos al controlador para hacer peticiones al feign que creamos:  
Texto

El contenido generado por IA puede ser incorrecto.

Luego lo corres y en eureka ya se agrego esto:  
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ahora si damos un método get al puerto 8082, veras que esta trayendo los datos del puerto 8081:  
[localhost:8082/mensaje-desde-ejemplo](http://localhost:8082/mensaje-desde-ejemplo)

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.