Zuul Routing

Zuul nos permite configurar un servidor de balanceo entre distintos servicios.

Se puede definir como un **proxy inverso o edge service** que nos va a permitir tanto **enrutar** y **filtrar** nuestras peticiones de manera dinámica, como monitorizar y securizar las mismas.

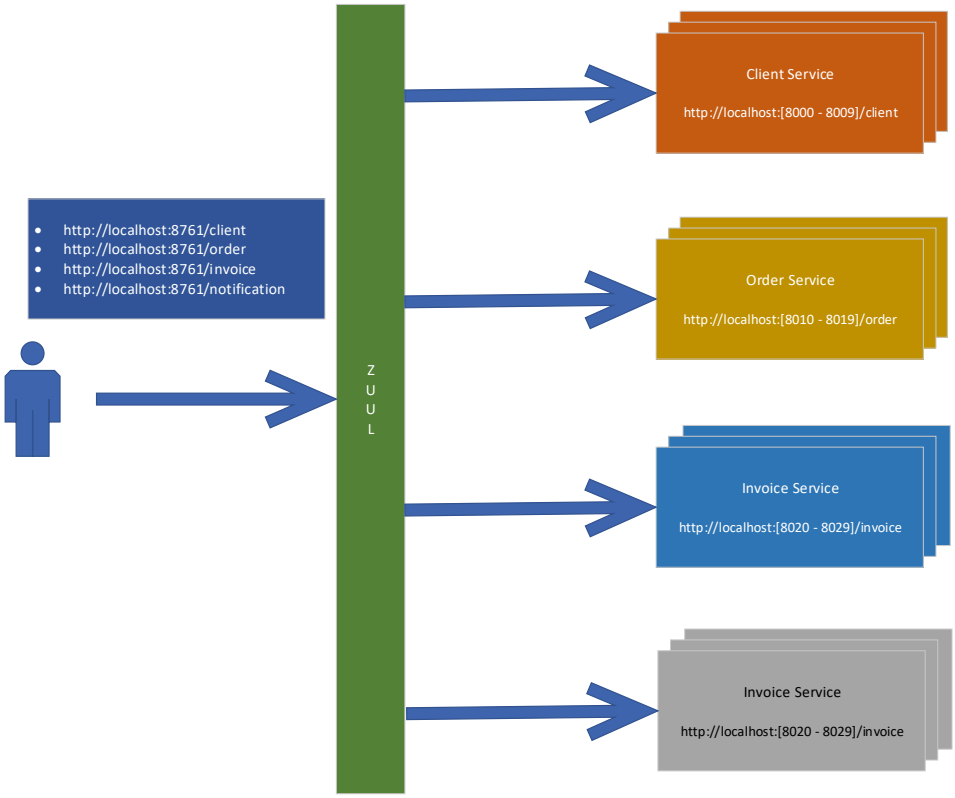
Hace uso de **Hystrix** (tolerancia a fallos) y **Ribbon** (balanceador de carga)

Como lo incluímos

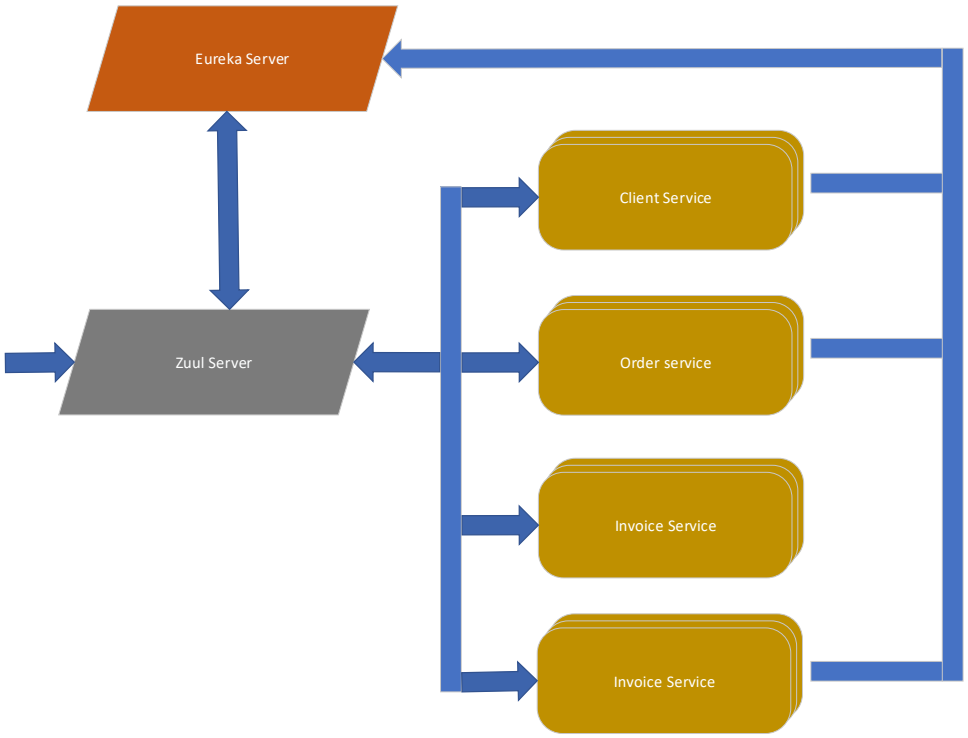
Tendremos que incluir las siguientes dependencias

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-zuul</artifactId>
    <version>2.2.2.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    <version>2.2.2.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-sleuth</artifactId>
    <version>2.2.2.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
    <version>2.2.6.RELEASE</version>
</dependency>
```

Arquitectura del ejemplo



Para esto tendremos



Propiedades

Ignorar servicios

```
ignored-services: '*'
```

Esta propiedad permite indicar que servicios queremos que zuul ignore. Por defecto, si no indicamos nada, zuul considerará toda la lista de los servicios de eureka. Si indicamos *****, estaremos diciendo que ignoramos todos los servicios de eureka, por lo que tendremos que agregar a mano los que nos interesan.

Eliminar Prefijo URI

```
stripPrefix: false
```

Por defecto, el prefijo es desmontado de la request antes de ser reenviada.

Esto NO tiene efectos en el prefijo que colocamos en el path de cada enrutamiento.

Esto lo que hace es eliminar parte de la URI a la hora de resolver el enrutamiento

Ejemplo

http://localhost:8762/client/endpoint1 -> http://localhost:8000/endpoint1

Enrrutamientos

Existen varias maneras de agregar enrutamientos.

```
zuul:
  routes:
    users: /myusers/**
```

Esta indica que todo lo que llegue a http://zuulserver/myusers/** se redirigirá al servicio **users** de la siguiente manera:

http://zuulserver/myusers/101 -> http://users/101

Cabe desatacar que zuul obtendrá la dirección del servicio users a través de eureka.

Sin embargo, podemos definir los enrutamientos de manera más certera

```
zuul:
  routes:
    users:
      path: /myusers/**
      serviceId: users_service
```

En este caso, **users** indica el nombre de la ruta, e irá a buscar a eureka el servicio **users-service**.

Se puede evitar hacer uso de eureka indicando la url

```
zuul:
  routes:
    users:
      path: /myusers/**
      url: https://example.com/users_service
```

Rutas y Spring Actuator

Para saber las rutas que posee el servidor zuul, necesitamos incluir la dependencia para agregar el actuator y luego indicar que queremos exponer los endpoints de esta librería.

```
management.endpoints.web.exposure.include: "*"
```

Si consultamos el **endpoint/actutor/routes** podremos ver las rutas que tiene definidas el servidor zuul.

Métricas

De la misma manera que con las rutas, podemos obtener las metricas de zuul en el endpoint **/actuator/metrics**.

Aquí nos saldrán todos los nombres de las metricas. Basta con acceder a la metrica que se quiere con **/actuator/metrics/nombre_metrica**.

Filtros

Se pueden implementar filtros en zuul para que trate las peticiones de acuerdo a lo que mas nos convenga.

Tipos de Filtro

- PRE: Se ejecutan antes de hacer la redirección.
- ROUTE: Se ejecutan después del pre y normalmente se usan para hacer conversiones de datos de request y response.
- POST: Se ejecutan antes de devolver la respuesta por parte de zuul.