

# Spring Cloud Bus y Rabbit MQ

En el curso vimos como podíamos crear un servidor de configuración donde teníamos la configuración de los microservicios.

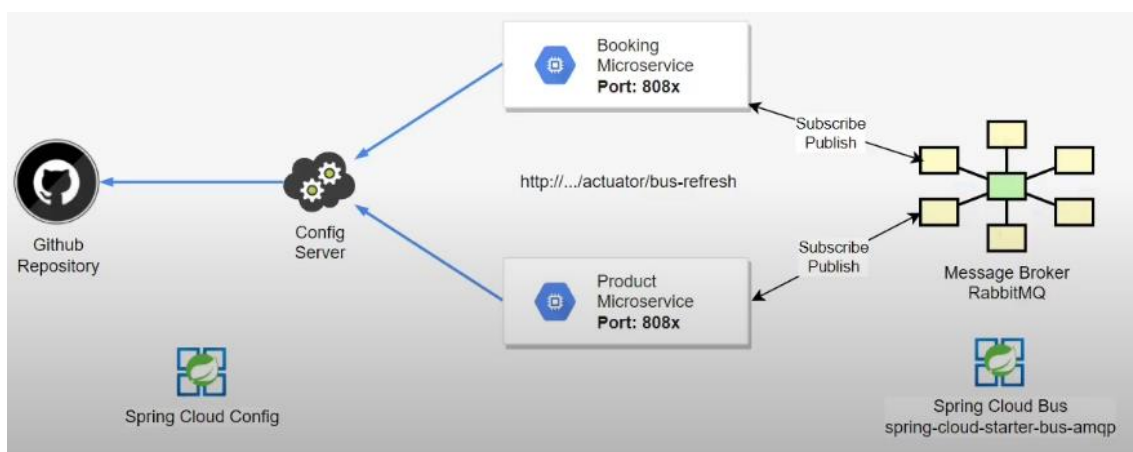
Recordamos que al levantar el microservicio, este obtiene su configuración del servidor de configuración y posteriormente se registra en Eureka Server.

Pero, que ocurría cuando modificamos la configuración y no queríamos reiniciar el microservicio? Pues que agregábamos la dependencia Actuator y utilizábamos la anotación `@RefreshScope` sobre el RestController.

El problema es que cada vez que hacemos un cambio en la configuración es necesario ejecutar:

POST `http://localhost:xxxxx/actuator/refresh`  
para que los microservicios cojan ese cambio.

Lo que pretendemos es no tener que hacer la actualización de forma manual.



Cada microservicio se suscribe al Message Broker y cuando haya cualquier modificación en el repositorio Github, el Message Broker publicara ese cambio y los microservicios leerán ese cambio para actualizarse.

Para implementar Message Broker se utiliza RabbitMQ

Se debe crear un nuevo micro-servicio con Spring Cloud Bus y cada microservicio otra dependencia Spring Cloud started bus amqp

Empezamos poniendo esta dependencia a cada micro-servicio que quiere obtener su configuración:

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-bus-amqp</artifactId>
  <version>3.0.3</version>
</dependency>
```

Ahora instalamos RabbitMQ, en su pagina web tenéis información de cómo hacerlo de múltiples formas: [www.rabbitmq.com](http://www.rabbitmq.com)

Una opción es hacerlo a través de Docker:

```
docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3.8-management
```

Una vez ejecutado este comando RabbitMQ estará escuchando en el puerto 5672.

Ahora debemos configurar los microservicios para que se conecten a RabbitMQ

En el archivo bootstrap.properties agregamos las siguientes propiedades:

```
spring.rabbitmq.host=localhost
spring.rabbitmq.port=5672
spring.rabbitmq.username=guest
spring.rabbitmq.password=guest
```

Levantamos los microservicios y realizamos la prueba de cambiar algo de configuración en el repositorio remoto de Github.

Para que todas las configuraciones de todos los microservicios se actualicen de forma global ejecutamos

POST <http://localhost:xxxx/actuator/busrefresh>

xxxx seria el Puerto de cualquier microservicio, pero no solo actualizara este sino todos.