

Rabbit MQ



¿Qué es?

- Es un bróker de mensajería es un programa intermediario entre aplicaciones que traduce los mensajes de un lenguaje a otro a través de un medio de telecomunicaciones. Permite comunicar los diferentes servicios que componen una aplicación entre sí. Es el intermediario entre los usuarios y los servicios a los que accede.
- Una analogía perfecta sería el sistema postal. Donde el emisor es el productor, el sistema postal es la cola y el receptor es el consumidor. El emisor (productor) se desentiende de la labor de enviar el mensaje y el receptor (consumidor) puede recogerlo de su buzón cuando le dé la gana, la bronca de hacer llegar el mensaje a su destino se la come el sistema postal (el bróker que implementa la cola).

Conceptos

- **bróker:** app que expone una serie de puertos a través de los cuáles podemos comunicarnos con él. Sin embargo para interactuar con dichos puertos se utiliza un Cliente. Un Cliente es una librería (en este caso para RabbitMQ) que abstrae cómo se insertan mensajes en la cola.
- **productor** (cliente en la analogía de los sockets): deja mensajes
- **consumidor** (servidor en la analogía de los sockets): lee mensajes (puede ser de una o varias colas)
- **cola de mensajes:** donde se dejan los mensajes
- **comunicación sincrónica:** es el proceso de hacer un get request para una API basada en web, ya que este se queda esperando una respuesta y no puede hacer nada hasta que la API responda.
- **comunicación asincrónica:** nos permite tener múltiples productores y consumidores.
- intercambios o exchanges (la lógica interna del servicio postal): concepto de los bróker de mensajería, se encargan de leer la metadata de los mensajes y realizar las acciones pertinentes según el escenario.
- **colas o queues** (los buzones personales de cada receptor): concepto de los bróker de mensajería, se encarga de ordenar los mensajes y son las estructuras receptoras de los mensajes procesados por la lógica de los intercambios.
- **vinculación o binding:** es lo que relaciona a los intercambios con las colas. una cola puede estar relacionada con múltiples intercambios y un intercambio a muchas colas. Incluso pueden haber colas no vinculadas a intercambios, por lo que no tendrán mensajes.

- **remote procedures call**: invocación remota de procedimiento
- **conexiones**: cada productor y consumidor establece una conexión tcp con el bróker. Una conexión, sin embargo, puede tener muchos canales.
- **canales**: mediante ellos un productor puede publicar múltiples mensajes usando hilos, todo esto dentro de una sola conexión. Lo mismo funciona para los consumidores. Abrir canales y no más conexiones ahorra un montón de recursos.

PD: los diferentes tipos de intercambios y vinculaciones de rabbitMQ nos permite una gran flexibilidad. Se pueden multiplicar mensajes pasándolos mediante un intercambio a múltiples colas o podemos configurar que solo una cola obtenga el mensaje de cierto intercambio.

¿Para qué sirve un bróker de mensajes?

- Permite minimizar el grado de conocimiento mutuo que las aplicaciones necesitan tener para intercambiar mensajes, implementando así un menor grado de interdependencia entre los módulos de un programa.
- Otorga al programador la habilidad de crear aplicaciones de acoplamiento flexible.
- es comúnmente utilizado en arquitecturas de micro servicios
- se independizan productores y consumidores
- Conseguimos invocación asíncrona, el mensaje se queda en la cola y libera al productor mientras que el consumidor lo procesa cuando pueda. En este sentido se diferencia de los sockets que solo cuentan con cliente-servidor y el servidor se queda esperando a una entrada de datos, lo que no le permite procesar otra petición.

características

- es asincrónico, es decir, no sigue un simple patrón de petición y respuesta.
- permite gran escalabilidad y fiabilidad.
- un bróker de mensajería tiene que soportar a muchos clientes, escribiendo en muchas colas a la vez y tiene que ser eficiente a la hora de procesar todos esos mensajes y enviarlos a los correspondientes consumidores. Por ello está implementado en earlang, que es un lenguaje de programación funcional que permite altos niveles de concurrencia.

implementación de cola de mensajes

- con docker desktop y rabbit MQ se puede correr el siguiente comando (con el nombre de versión de rabbitMQ correspondiente): `docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3.9-management`

- esto permitiría que al abrir el navegador y colocar la dirección: "localhost: 15672" podamos usar rabbit mq - Sin embargo nos pidió una instalación particular de rabbitMQ, elastic search y kibana por proyecto.

Bibliografía

- [Why rabbitMQ?](#)

- [colas de mensajes - Patxi Gortázar parte1](#)
- [colas de mensajes - Patxi Gortázar parte 2](#)
- [¿Qué es un bróker de mensajería?](#)
- [core concepts](#)