

`durable` 이란 broker가 재시작되어도 queue가 삭제되지 않고 남을 수 있게 만드는 속성이다.

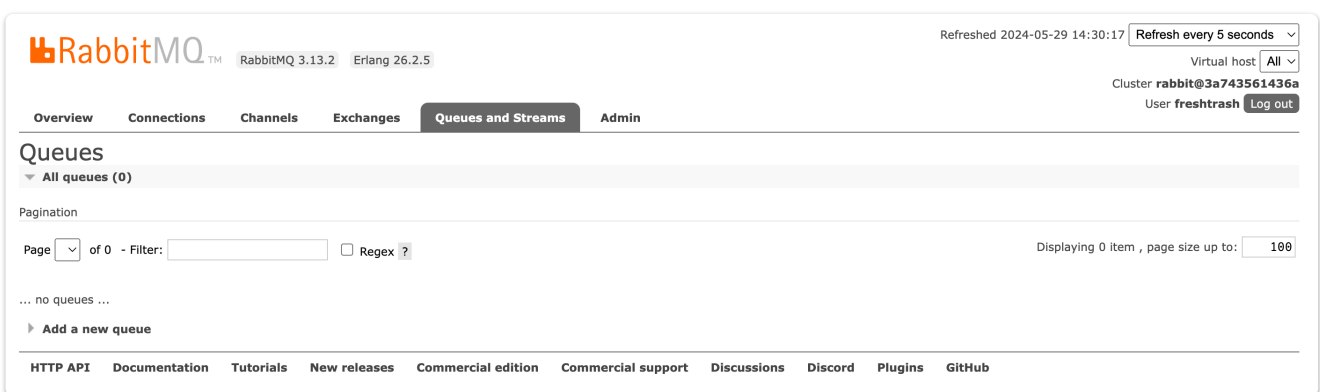
(RabbitMQ를 Docker 컨테이너로 올려 실행했다.)

Non-Durable

`durable` 을 `false`로 설정할 경우 broker가 재시작되면 queue가 삭제될 것이다. 실제로 그런지 한 번 확인해보았다.

Broker 실행

broker를 처음 실행하면 아래처럼 queue가 아무것도 없을 것이다.



백엔드 서버 실행

Spring Boot로 개발한 백엔드 애플리케이션을 실행하게되면 다음과 같이 설정한대로 queue가 생성된다.

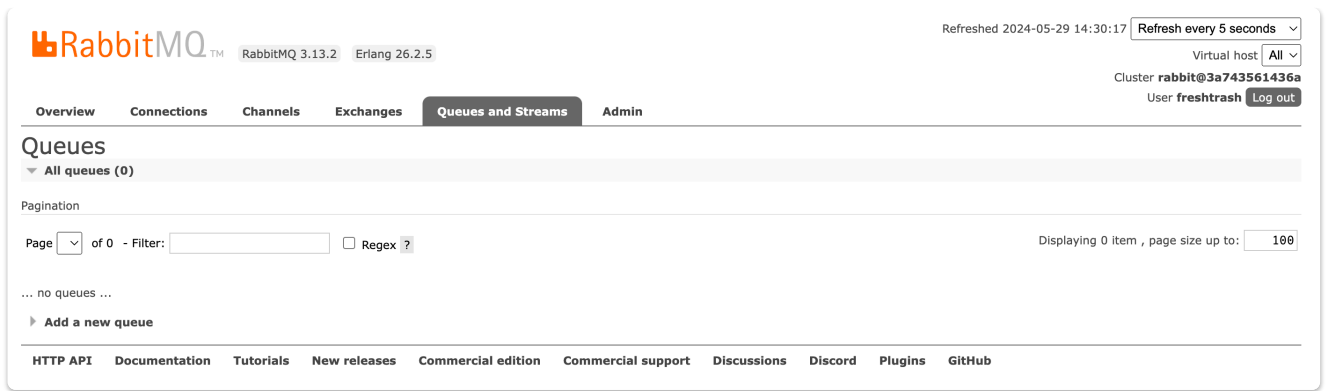
Overview					Messages			Message rates		
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
/	queue.chat	classic		running	0	0	0			
/	queue.waste.changeStatus	classic		running	0	0	0			
/	queue.waste.complete	classic		running	0	0	0			
/	queue.waste.flag	classic		running	0	0	0			

Broker 재실행

broker가 재실행할 때 두 가지 경우를 생각해볼 수 있다.

1. 백엔드 서버가 실행되는 중 Broker가 재실행되는 경우
2. 백엔드 서버가 다운되었을 때 Broker가 재실행되는 경우

지금의 경우 broker가 재실행되었을 때 queue가 삭제되는지 확인해야하기 때문에 두 번째 경우를 확인해 보았다.



결과는 예상대로 queue가 모두 삭제되었다.

하지만 첫 번째 경우(백엔드 서버가 실행 상태인 경우) broker가 재실행되면 queue가 모두 삭제되었다가 다시 생성된다.

Durable

반대로 `durable` 을 `true`로 설정하면 broker가 재실행되어도 삭제되지 않아야한다.

백엔드 서버 실행

앞서 `durable` 을 `false`로 설정한 것과 다르게 Features라는 곳에 `D` 라는 것이 추가된 것을 확인할 수 있었다.

Overview					Messages			Message rates		
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
/	queue.chat	classic	D	running	?	?	?			
/	queue.waste.changeStatus	classic	D	running	?	?	?			
/	queue.waste.complete	classic	D	running	?	?	?			
/	queue.waste.flag	classic	D	running	?	?	?			

마우스를 대보면 `durable: true` 표시라는 것을 알 수 있다.

Features	State
D	running
D	running
D	running
D	running

Broker 재실행

그리고 broker를 재실행했을 때 다음과 같이 queue가 삭제되지 않고 그대로 남은 것을 확인할 수 있었다.

Overview					Messages			Message rates		
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
/	queue.chat	classic	D	running	?	?	?			
/	queue.waste.changeStatus	classic	D	running	?	?	?			
/	queue.waste.complete	classic	D	running	?	?	?			
/	queue.waste.flag	classic	D	running	?	?	?			

Spring AMQP

durable 설정

Spring AMQP를 사용하여 빈을 등록할 경우 다음과 같이 Queue를 생성할 때 durable을 false/true로 지정해주면 된다. (default이 true)

```
new Queue("testQueue", false);
```