

## 안전하게 Publish하기 위한 설정

메시지는 메모리에 저장되며 재실행되거나 라우팅할 수 없는 경우 유실될 수 있다.

1. `mandatory` 를 `true`로 설정해주면 `queue`에 메시지가 추가되었을 경우 `success`를 반환하고 라우팅되지 않을 경우 `error`를 반환한다.
2. `persistent` 를 `true`로 설정해주면 메시지가 `disk`에 저장된다. (동시에 `RAM`에 저장됨)
3. `open channel`에서 `waitForConfirmsOrDie` 을 호출하여 메시지가 정상적으로 확인되었는지 알 수 있다. `Spring AMQP`에서는 `ReturnsCallback` 또는 `ConfirmCallback` 을 사용할 수 있다.

또한 `lazy queue mode`를 사용하여 가능한 빨리 `disk`로 옮기도록 할 수 있다. 이렇게하면 더 적게 메시지를 `RAM`에 유지할 수 있게 만든다. (java에서 args에 `x-queue-mode=lazy` 를 추가해주면 된다.)

### Important

As of RabbitMQ 3.12 the queue mode is ignored and classic queues behave in a similar manner to lazy queues.

## 코드 적용

### mandatory

`mandatory` 설정은 다음과 같이 `rabbitTemplate` 빈을 등록할 때 `true` 로 값을 지정해주면 된다. 그리고 `ReturnsCallback` 을 정의해주면 라우팅되지 못한 경우 처리해줄 수 있다. 아래 예시에서는 간단하게 `log`를 출력하도록 했지만 실제로는 `DLQ`로 보내도록 구현하면 될 것 같다.

```
@Bean
RabbitTemplate rabbitTemplate(ConnectionFactory connectionFactory) {
    RabbitTemplate rabbitTemplate = new RabbitTemplate(connectionFactory);
    // mandatory = true
    rabbitTemplate.setMandatory(true);
    // 메시지가 브로커에 도착했지만 지정된 큐로 라우팅되지 못한 경우
    rabbitTemplate.setReturnsCallback((returnedMessage) -> {
        log.info("routingKey: {}, replyText: {}",
            returnedMessage.getRoutingKey(), returnedMessage.getReplyText());
    });
    return rabbitTemplate;
}
```

추가로 `publisher-returns` 를 `true`로 지정해주어야한다.

```
spring:
  rabbitmq:
```

```
...  
publisher-returns: true
```

## persistent

Spring AMQP에서는 `MessageProperties` 을 확인해보면 기본적으로 `persistent`가 설정되어있다.

[reference]

- <https://blog.devgenius.io/guaranteed-message-delivery-with-rabbitmq-5211cff5f1e3>
- <https://www.rabbitmq.com/tutorials/tutorial-seven-java>
- <https://www.rabbitmq.com/docs/lazy-queues>
- <https://www.rabbitmq.com/docs/classic-queues>
- <https://www.rabbitmq.com/blog/2023/05/17/rabbitmq-3.12-performance-improvements>
- <https://docs.spring.io/spring-amqp/reference/amqp/template.html#template-confirms>