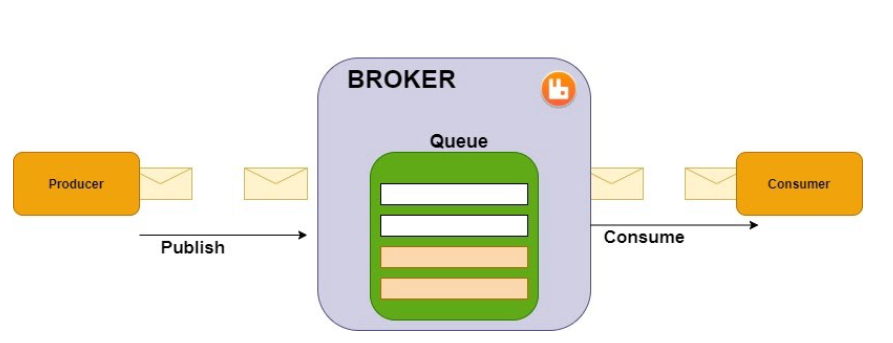
-message broker implements Advanced Message Queuing Protocol



**1.Producer:** publishes a message to an exchange.

**2.Message:** info/data

3.**Queue(FIFO Data stuct):**  holds many messages. This is where the broker works

4. Customer: The system that receives the message from the broker

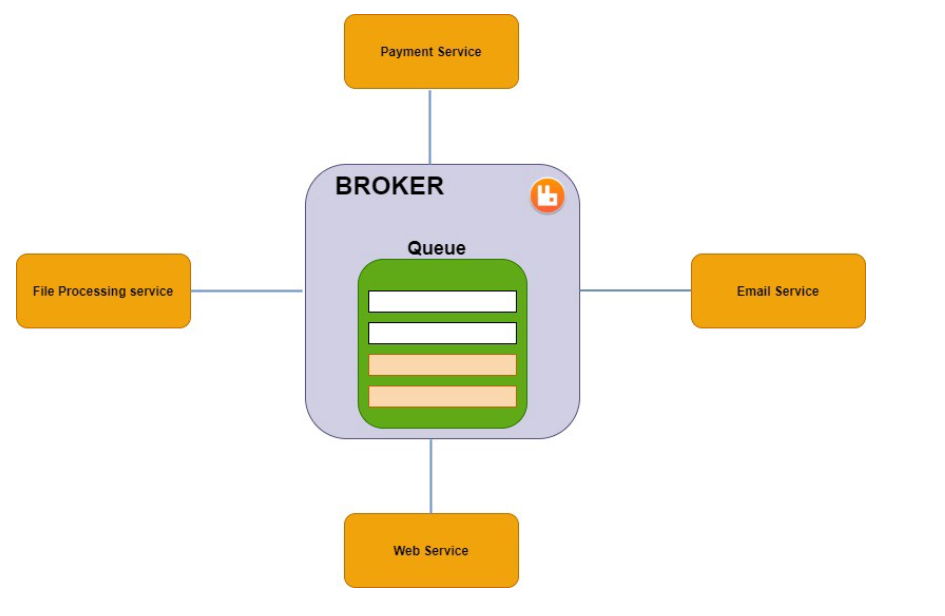


Figure : Demo service

Installing Rabbit MQ(in docccker image):

docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management

By default, RabbitMQ creates a guest user.

Login: guest

Password: guest

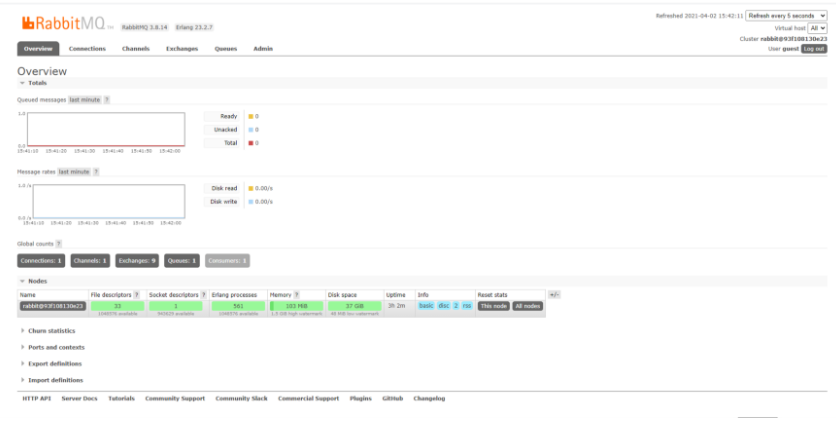


Fig : after login

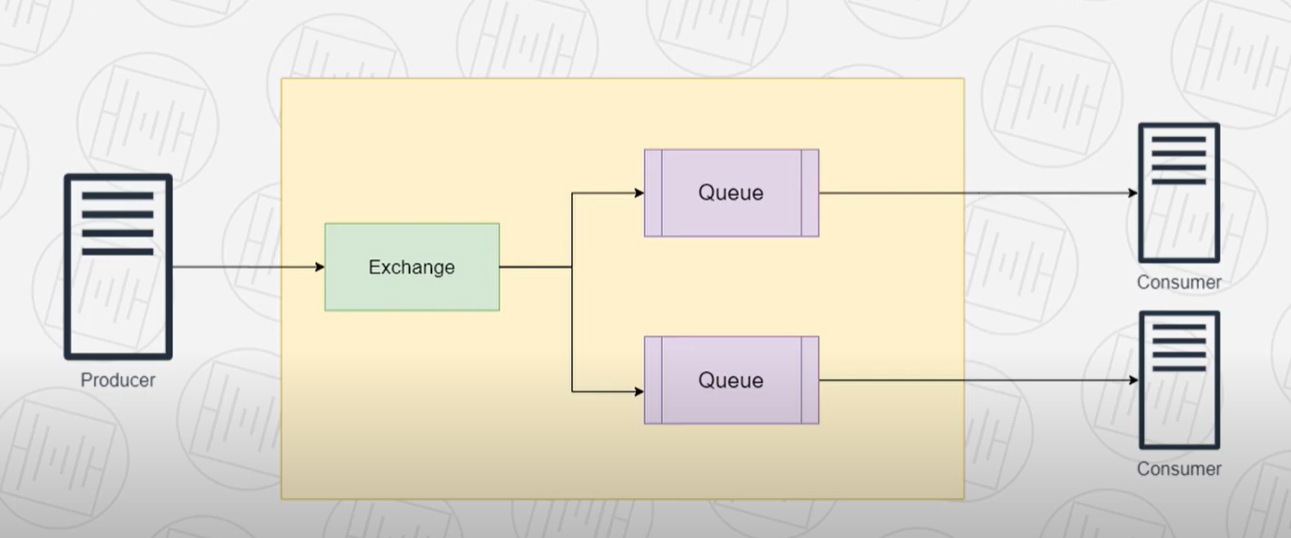
**Integrate Rabbit MQ with spring boot**

1. Required dependencies :

-Spring Boot DevTools

-Spring for RabbitMQ

1. Now let’s create a config package and inside it create RabbitMqConfig class where the beans must be defined.
2. RabbitMqService must be created.
3. RestController must be created to message from rabbit MQ
4. Sending messages through the controller will be received in <http://localhost:15672/#/queues>(rabbit mq service)



Rabbit MQ doesnt send messages directly, instead sends it to exchange

https://hamdi-bouallegue.medium.com/integrate-the-message-broker-rabbitmq-with-spring-boot-ee94ea489706