To Start rabbit mq type below in command

C:\Users\Rkumar13\dev\tools\RabbitMQ Server\rabbitmq\_server-3.8.1\sbin

rabbitmq-server

if started try to browse below url in your browser

<http://localhost:15672/#/>

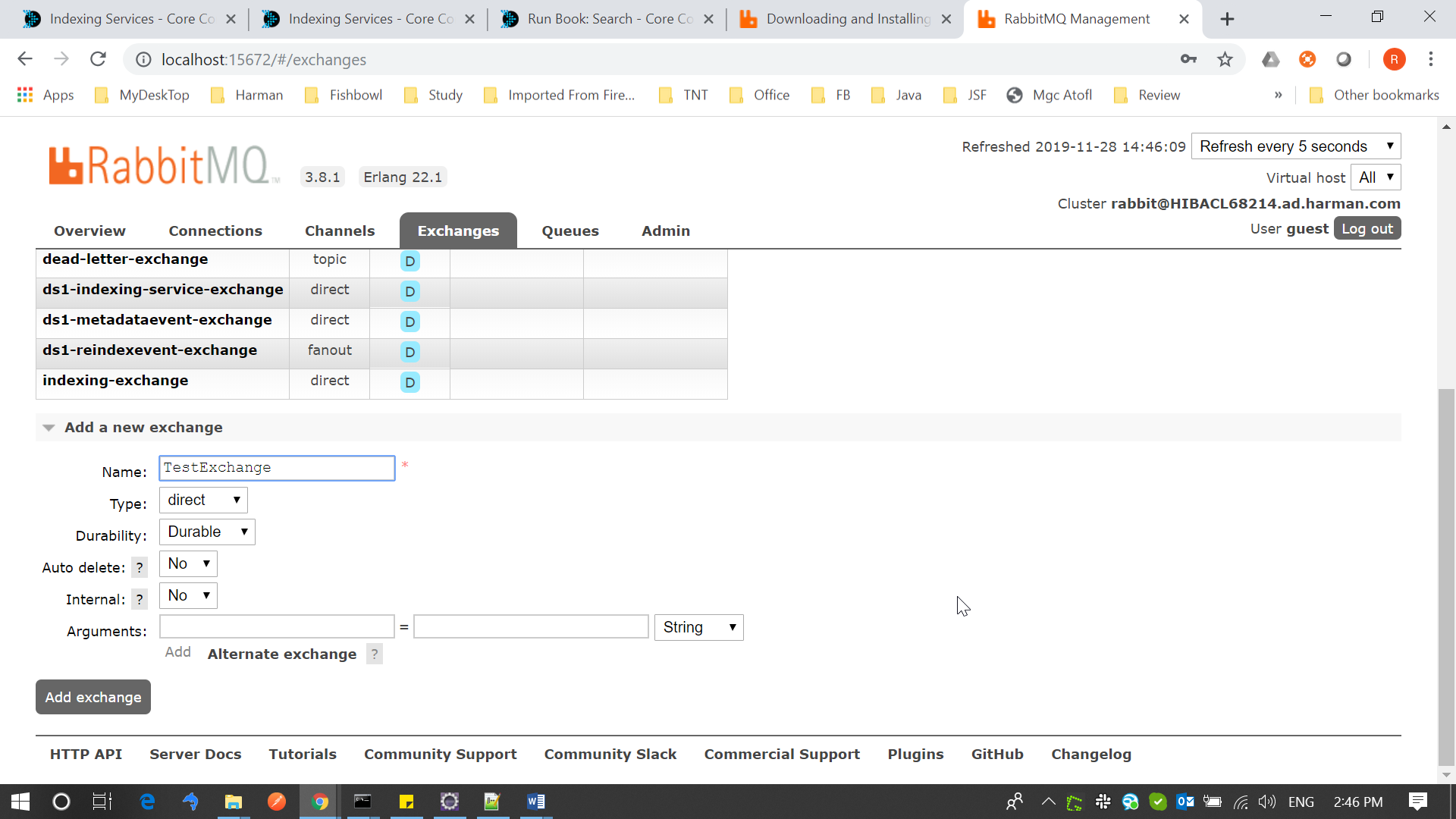
if above url is not working then use below command to bring up the RabbitMQ management service

rabbitmq-plugins enable rabbitmq\_management

The default id password is guest/guest

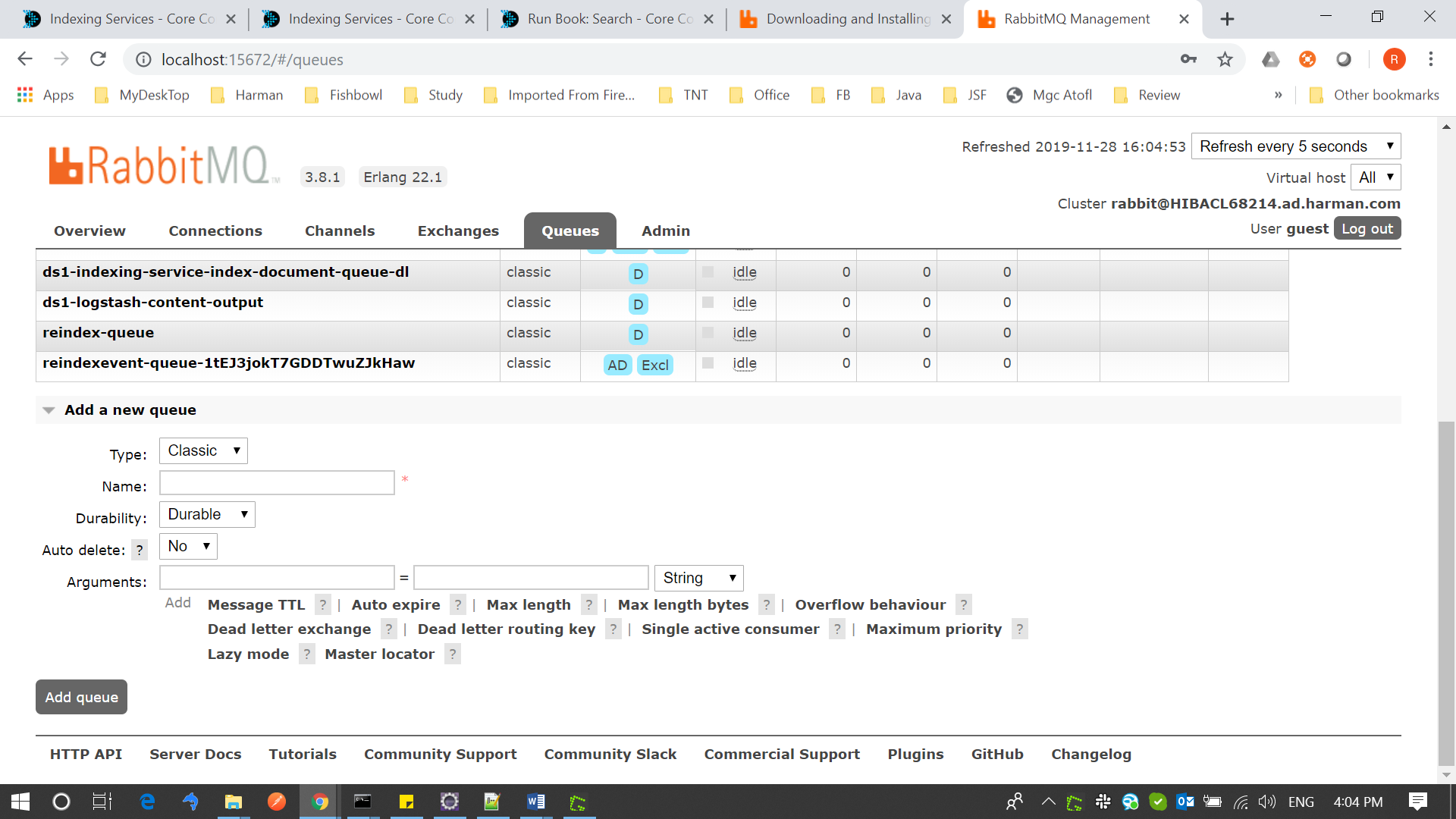
**Exchanges**

* Go to exchange tab and create new exchange by clicking on add a new exchange.

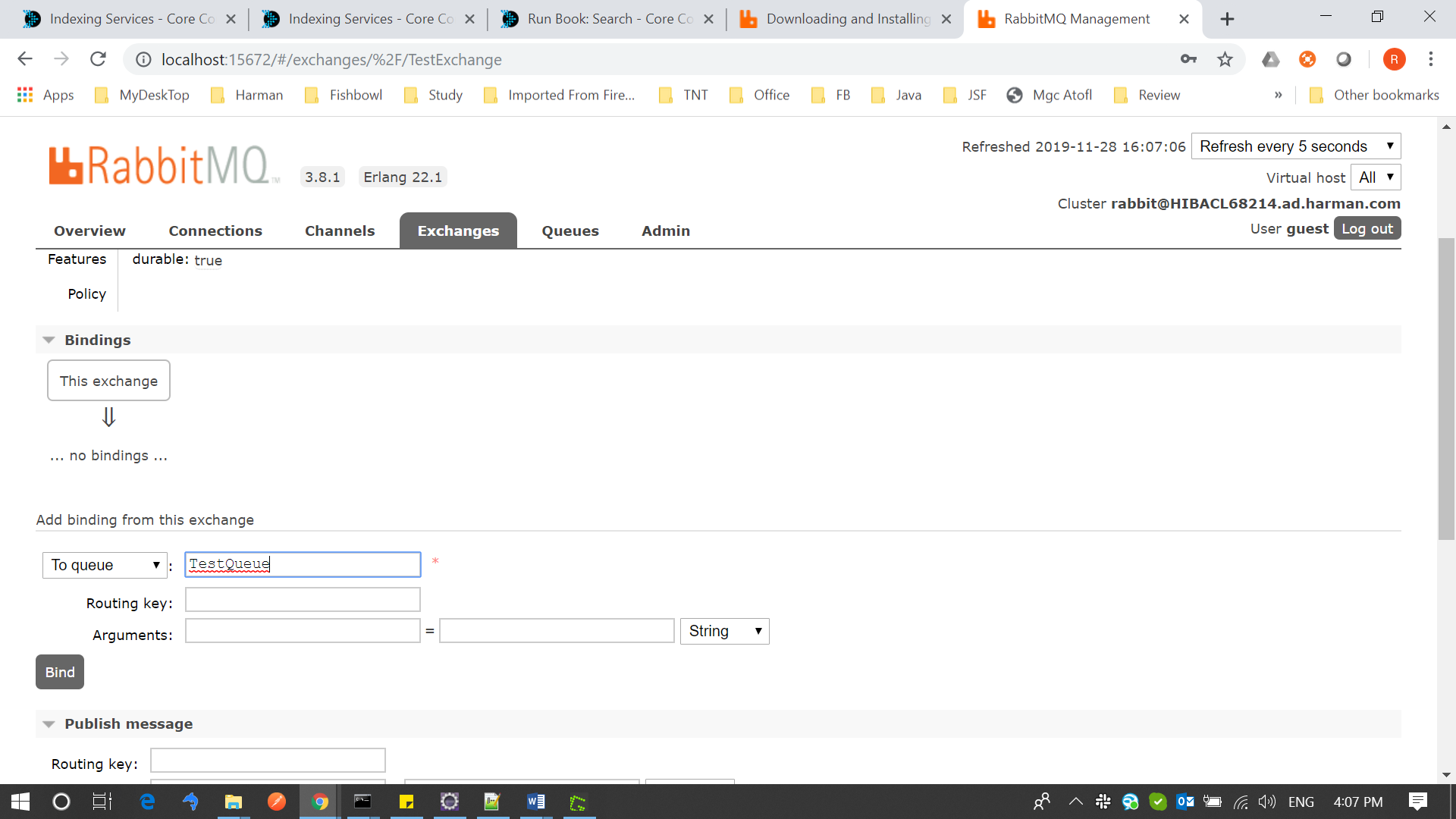


**Queue**

To create the new queue, navigate to queue section create the queue



Now go back to exchange and bind this queue to exchange.

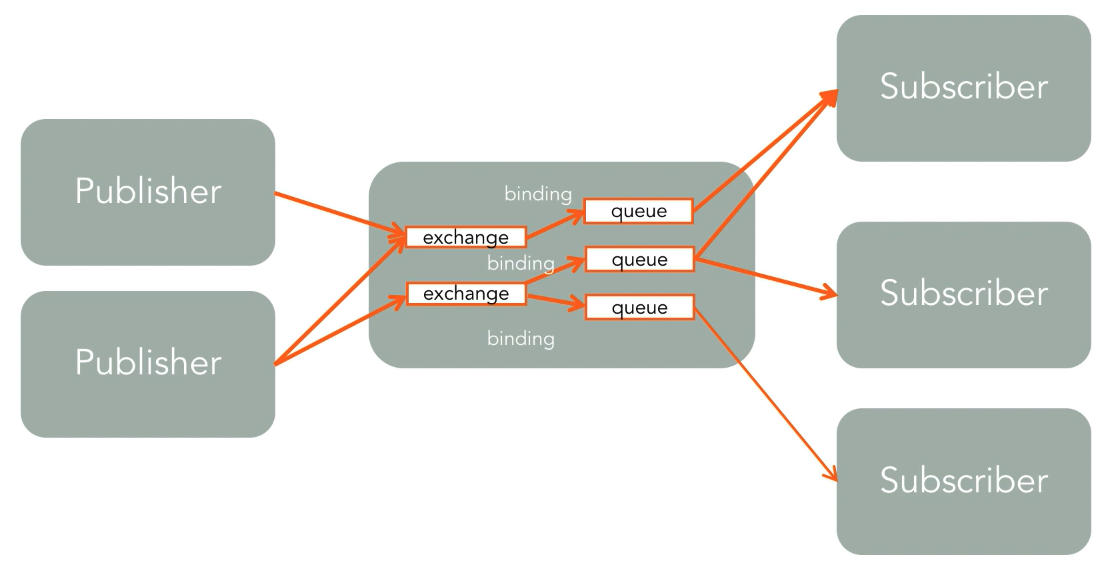


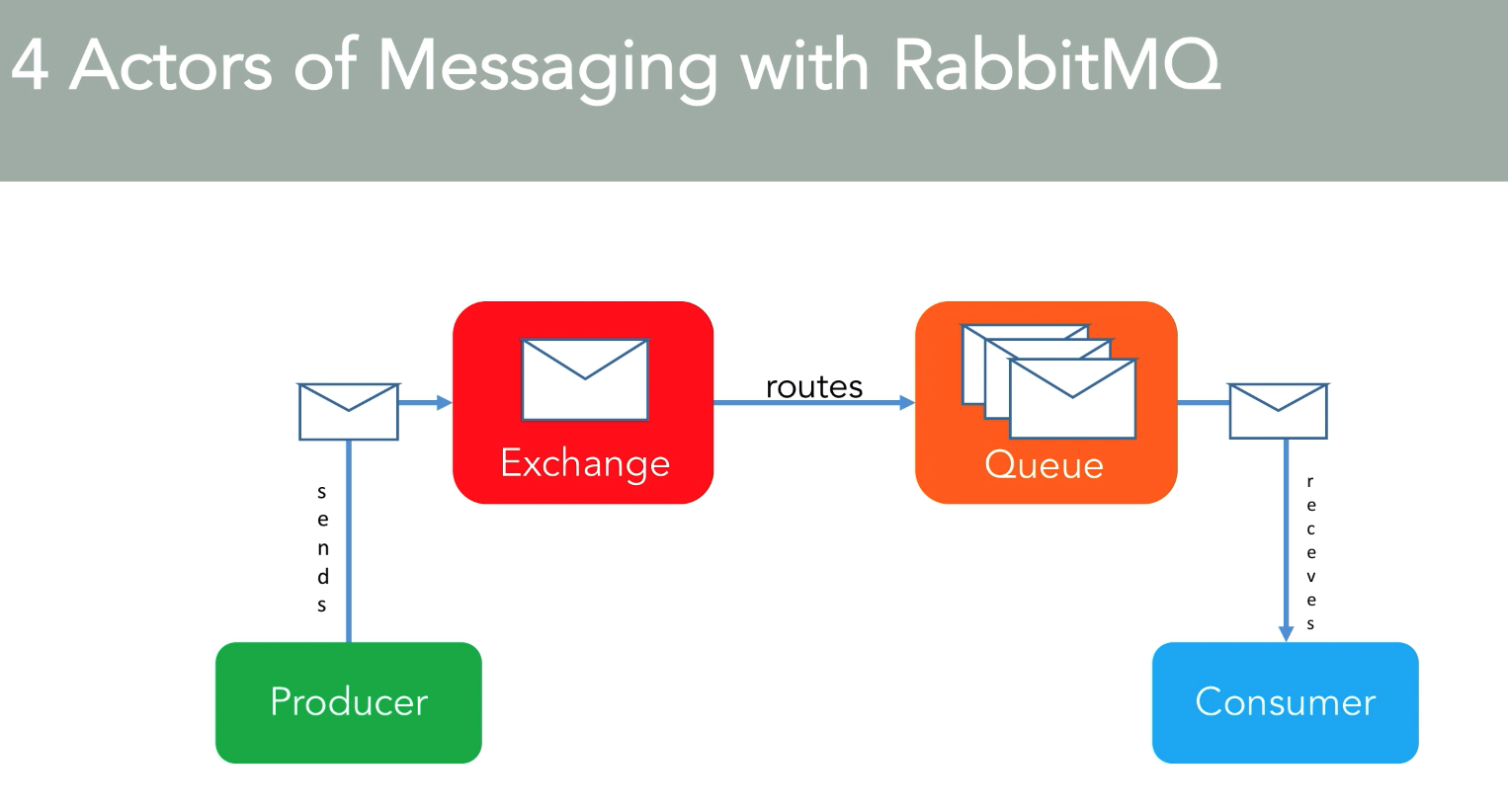
AMQP

Advance Messaging Queuing Protocol

**RabbitMQ**

* RabbitMQ is developed on Erlang programing language**.**
* Allows multiple connection channels inside a single TCP connection in order to remove the overhead of opening a large number of TCP connections to the message broker.





**Exchanges**

* Actual AMQP elements where messages are sent first.
* Takes a message and routes it into one or more queues.
* Routing algorithm decides where to send messages from exchange.
* Routing algorithms depends on the exchanges types and rules called “bindings”, these bindings are the actual configurations when you bind the exchanges then your message first comes to the queue and then it comes to the queue throw these bindings, which keeps the things together between the exchange and queue.
* Bindings are simply used to bind exchanges to queues for the message delivery.

Four exchanges types :

Default pre-declared names

1. Direct Exchange : (Empty string) and amq.direct : This is the default exchange, this is the place where messages are get delivered when there is no bindings or configurations.
2. Fanout Exchange : amq.fanout : This is used as message distribution to everywhere
3. Topic Exchange : amq.topic : Topic exchange is work on the topic which is pre defined.
4. Headers Exchange : amq.match (and amq.headers in RabbitMQ) : it’s a way to exchanges headers in RabbitMQ

**Queues**

* A core elements in any MQ protocol especially for RabbitMQ.
* Messages are routed to queues from exchanges.
* Queues are final destination in RabbitMQ before being received by subscribers, because consumer doesnot knows about the exchange and it only communicate with the queue.
* Routing algorithms depends on the exchange type and rules called “bindings”.
* Bindings are simply used to bind exchanges to queues for message delivery.

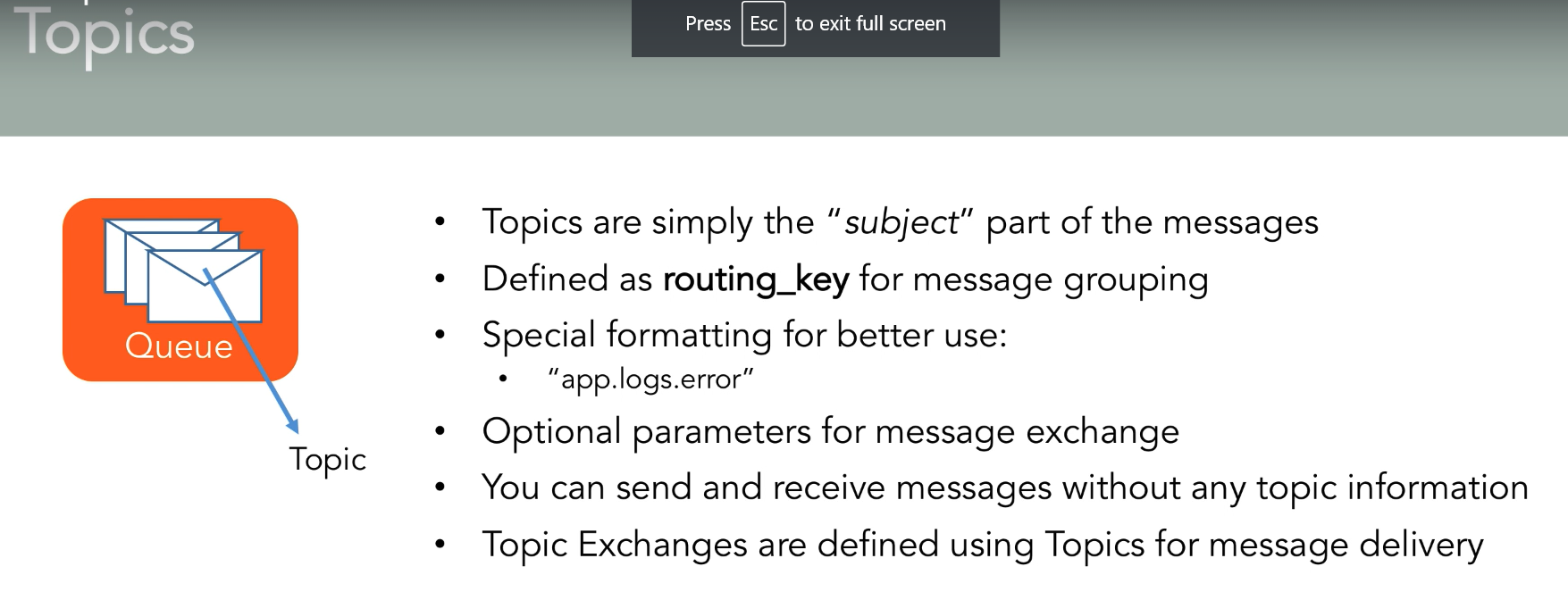
Property of Queue:

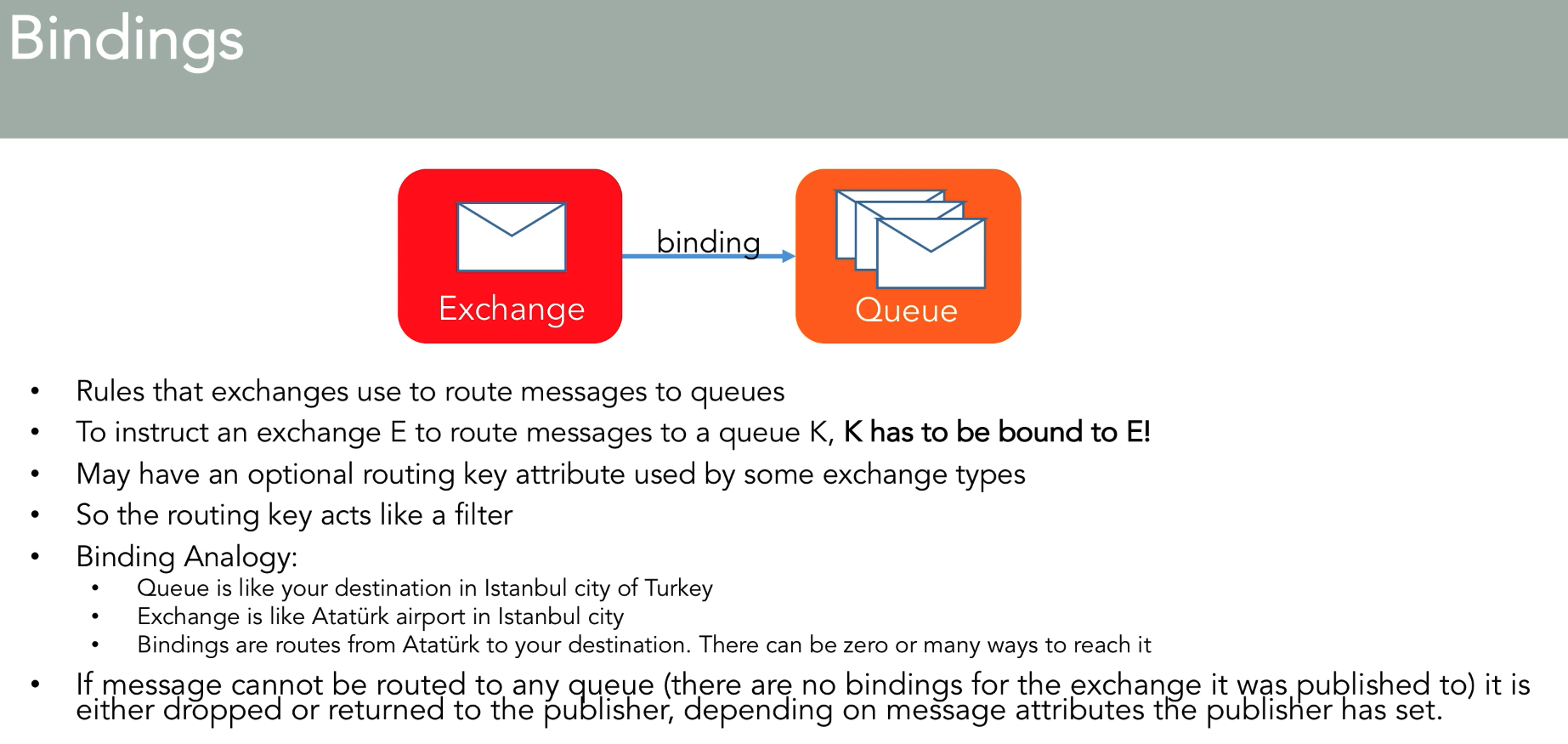
Name : The name of the queue.

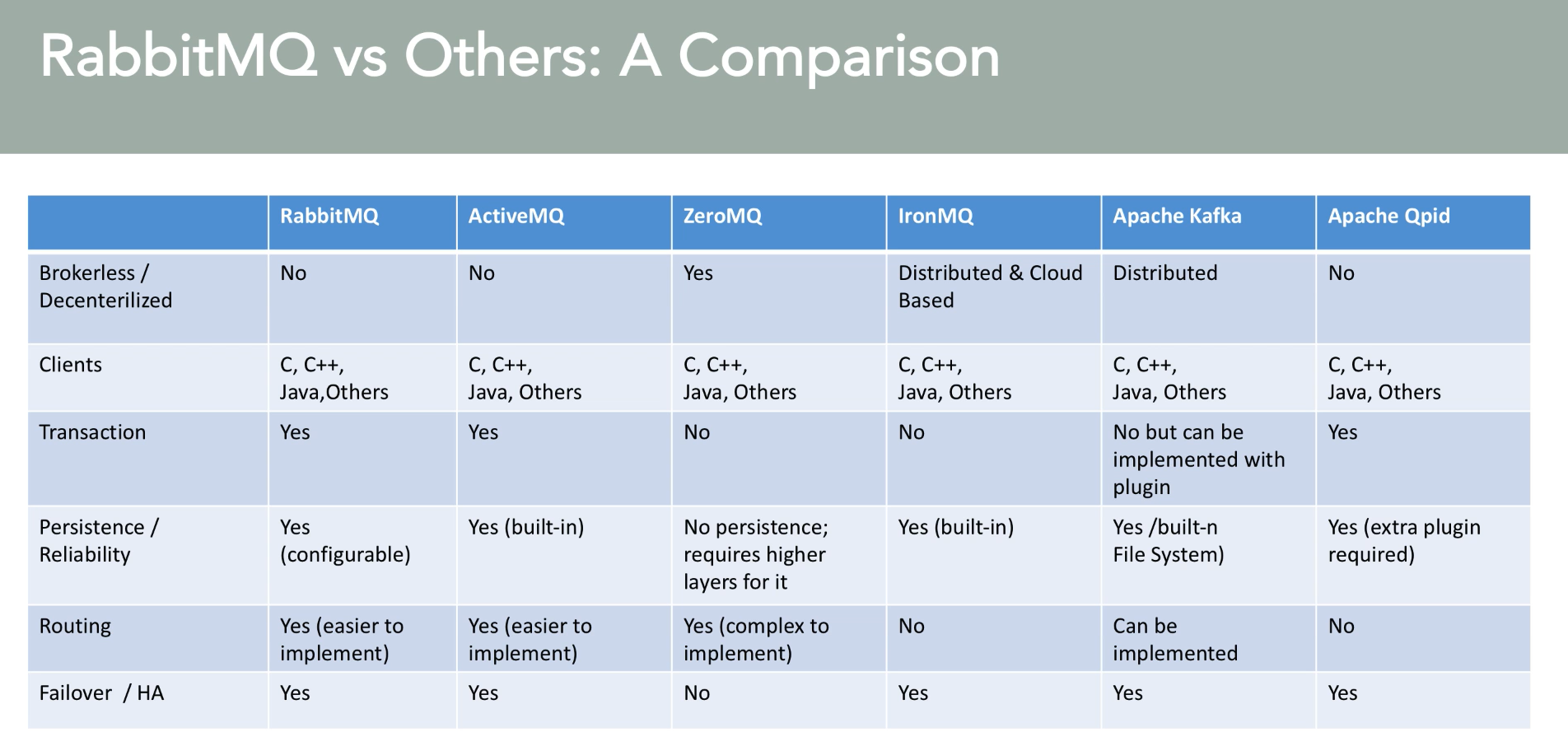
Durable : Either persist to queue to disk or not, there is durable queue also in exchange.

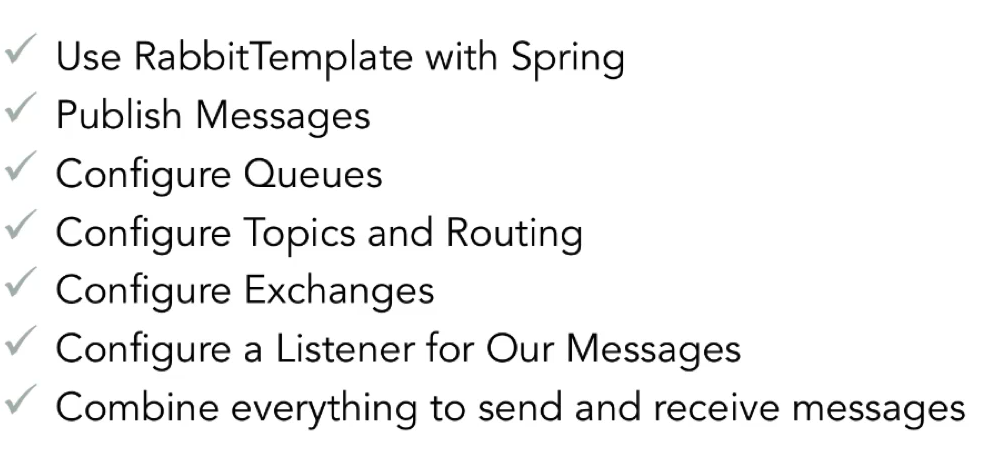
Exclusive : Delete the queue if not used anymore, if there is no data in queue then it will get deleted.

Auto-Delete : Delete the queue when consumer unsubscribe. Queue can be deleted when there is no subscribers in the queue





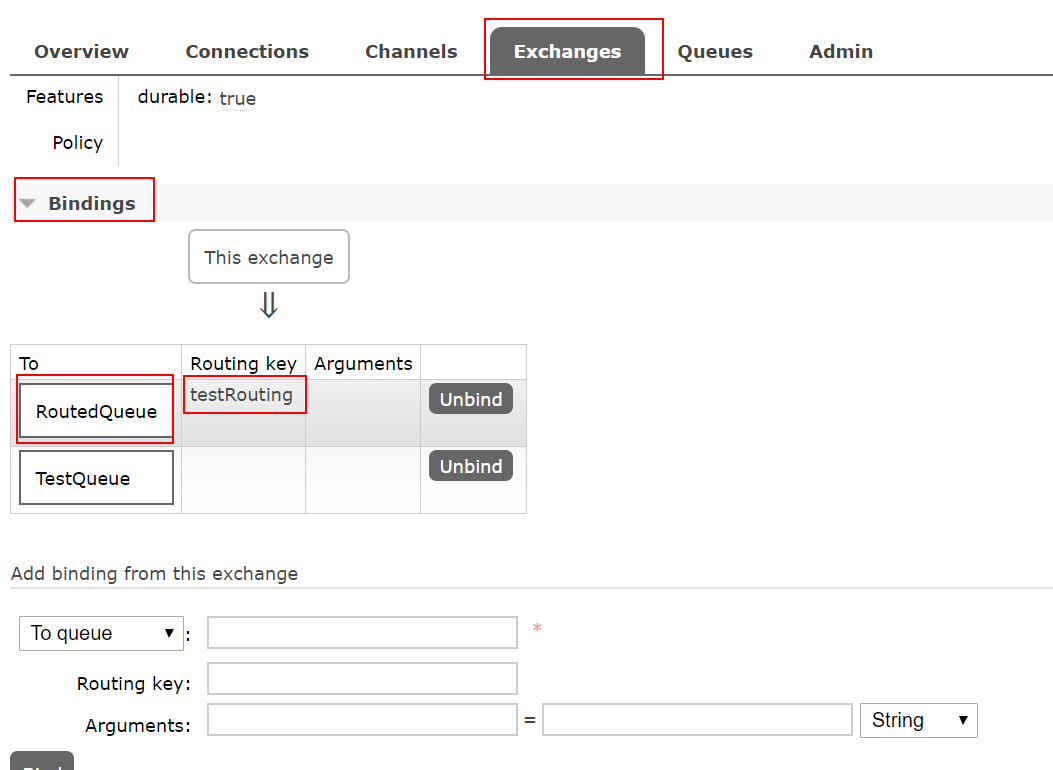




**Code Practice**

How to send the message to queue





**Publishing binary message with RabbitTemplate**

For publishing binary message the pojo class should be implements **Serializable** else it will result in exception – that only Serializable class are allowed.

**RabbitMQ with SpringBoot**

<https://www.javainuse.com/messaging/rabbitmq/listeners>

**durable & non-durable queue**

Do not misunderstand a non durable queue to be a temporary queue. Durability property is related to how long a message will be stored in the queue. For example for in RabbitMQ restart messages in non durable queue will be lost while those in durable queue will not be lost.