

RabbitMQ Notes.docs

Its an open source message queue

In this project we used Allow asynchronous messaging with java and spring boot

You can say rabbitMQ is the message broker

Which allows two or more software to talk to each other with the help of messages

Different message protocols are used for ex--

MQTT

AMQP(protocol we are gonna focus on)

A- aynchronous

M- messaging

Q- queue

P- protocol

How to start rabbitMQ in the server

```
sh rabbitmq -server
```

To watch management plugin in the browser

Type following command in terminal/shell

```
sh rabbitmq-plugin enable management-plugin
```

Default password are and username are guest

Address is- <http://localhost:15672/>

Q- what is messaging and why do we need it?

Ans- A message contains a header and a body

A message is the data that is passed or we can call exchanges between two or more applications

The passing of messages is called messaging

For Example---

Method calls

Remote Procedure calls

Http Clients and Server

But most of this messaging is serial synchronous and tightly coupled

But messaging should be loosely coupled or we can say asynchronous

Q- but why do we need asynchronous loosely coupled messaging system?

Ans- imagine you are getting thousands of orders request from customers to an ecommerce site server all at once

Then we might crash the server with all the load to process all the request

So here the progressively loosely coupled asynchronous messaging helped

We make a queue of all these messages and process them one by one..

We use messaging protocol To achieve this

Advance structure is provided like Queue Topics Channels Exchanges.

There are mostly 3 message protocols

1>>STOMP(Simple Test Oriented Protocol)

- > provides an interoperable format so that stomp clients communicate with the stomp broker to provide easy and widespread messaging.

- > very simple and easy to implement http like design

- > use “send” semantics with “destination” string for where message to deliver

- > consumer of the messages subscribes to the destination

2>>MQTT(Message Query Telemetry Transport)

- > supports 100% of Concurrent Devices Support

- > its lightweight contains compact binary packets,no message properties,compressed headers,

- > support machine to machine(Internet Of Things) connectivity protocol.
- > highly standardize
- > mainly made for resource constraint device and low band width.
- >high latency network dial up lines and satellite link
- > used for full feature enterprise messaging
- > AWS IOT and Green Grass Edge computing solutions are built around MQTT

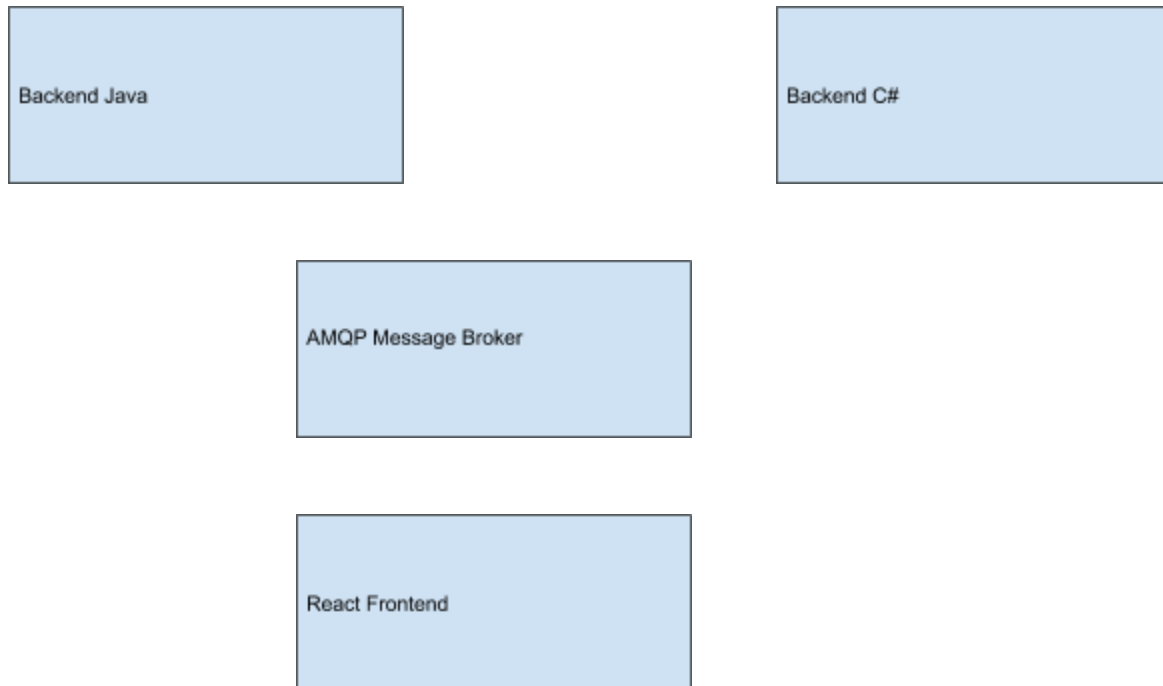
>>>>>>> AMQP<<<<<<<<
Is What WE Use In RabbitMQ

Its reliable and interportable
Advance
Message
Queue
Protocol

- > provides a wide range of features related to the messaging
- > topic based publish and subscribe messaging
- > Flexible Routing ,Transaction & Routing

AMQP in detail

Main aim -- helps in communicating between two completely different systems.



>All amqp clients are interportable with all amqp server

> diverse programming language can communicate easily

>Enable messaging as a cloud service

> Transactional messaging functionality

Advantages Of AMQP

> reel time feed of constantly updated information

> want encrypted assured transaction

> want to deliver message when destination comes online

> send enormous messages while receiving messages while receiving the messages while receiving status updates over the same network communication

> work on all os support all languages

RABBIT MQ

- >> Its an open source message broker
- >> message oriented middleware
- >> popular implementation of amqp
- >> provide a robust and flexible messaging platform
- >> can work with others.
- >> Develop using erlang programming language
- >> high throughput and low latency
- >> Support clustering fault tolerance and scalability
- >> Allow multiple connection channel inside a single tcp connection in order to remove the overhead of opening large number of tcp connection

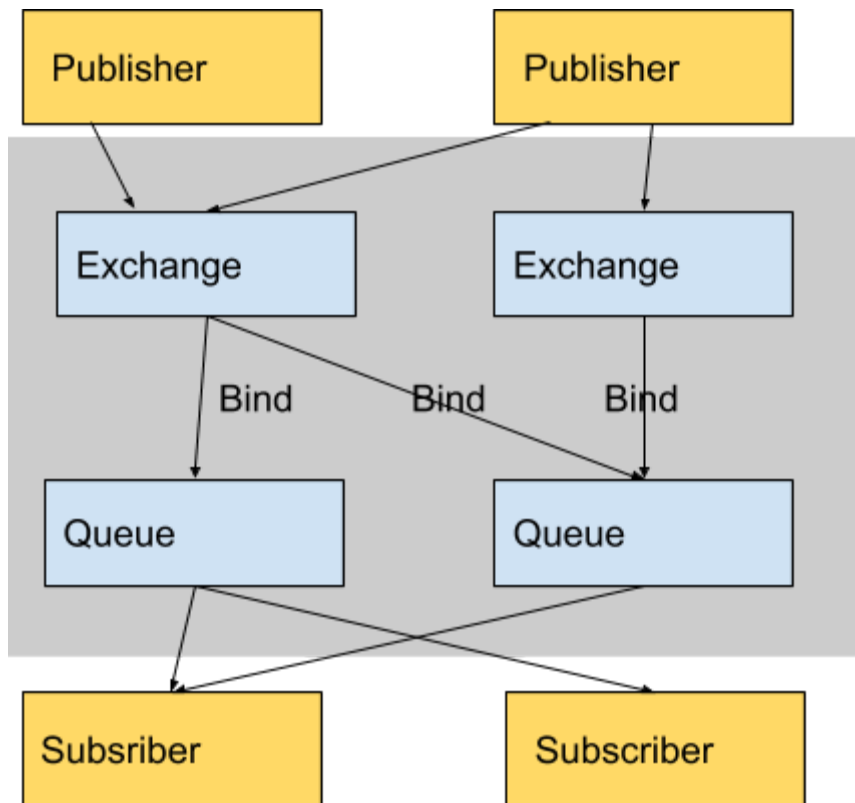
1> Exchanges in rabbitMQ

Exchanges can be default or custom

Exchanges are real amqp element

Takes messages and route it to one or more queue

Exchanges contain



Rabbit MQ architecture

- > publisher is connected to exchange
- > exchange binded to queue
- > subscriber can be listening to multiple queue
- > Routing algorithm also called binding decides where to send messages from that exchange.
- > exchanges are bindided to single or multiple queue

4 type of exchanges

1> Direct Exchange

Also called as default exchange

Amq.direct

Messages go to direct exchange if they have no binding or configuration to message or queue

2> FANOUT Exchange

By default is called `amq.fanout`

Messages are being spread out to all the queues binded to that exchange.

3> Topic Exchange `amq.topic` (default exchange it creates)

It has a specific way of delivering the message.

You define the topic and messages are sent to only that topic.

4> Headers Exchange

You want to exchange headers between producers and consumers

Queue

Subscribers only know about the queue and nothing about exchange. queue is the final destination in rabbit mq before being subscribed by subscriber..

Binding tells routing algorithm between exchange and queue

Properties of the queue

- 1.Queue has a name
- 2.Its durable ie we should persist queue to disk or not
- 3.Exclusive ie if items in queue can be deleted or not

4.Auto-Delete When queue has no messages and is not subscribed to any consumer it deletes automatically

routingKey are part message

They are defined as routingKey for messaging group

With routingKey you can deliver the message to a particular queue and subscriber

For example if queue are the destination say D
And publisher act as starting point say p
Then path from P to E will be called as Routing algorithm or Binding

If there is no binding between exchange and queue the message is ie dropped or returned to publisher