RabbitMQ

Protocollo non avente supporto nativo per JMS basato su AMQP (Advanced Message Queuing Protocol). E’ scritto in Erlang e può essere installato attraverso [Chocolatey](https://chocolatey.org/install) o manualmente installando entrambi i pacchetti (Erlang e RabbitMQ) in format .msi o .exe.

Se si decide di utilizzare Chocolatey, dopo aver seguito i passi per la sua installazione al percorso linkato sopra, si riavvii la powershell e si esegua il comando:

**choco install rabbitmq**

RabbitMQ è un **MessageBroker** completo e non è necessario il supporto di altri broker come apache artemis o altri message-broker noti. Va solo importata la relativa dipendenza:

**<dependency>**

**<groupId>org.springframework.boot</groupId>**

**<artifactId>spring-boot-starter-amqp</artifactId>**

**</dependency>**

E le note librerie Jackson per il binding degli oggetti.

Una volta installato sarà possibile **avviare** e **stoppare** il servizio cercando **RabbitMQ** nella pannello di ricerca di Windows. Per recarsi alla console aprire una pagina web e recarsi all’indirizzo: <http://localhost:15672/>

user: guest

password: guest

Principali Entità: Exchanges, Binding, Queue

Queste parole chiave sono anche conosciute come entità AMQP:

* **Exchange (Scambi)**: Gli *Exchange* sono entità a cui vengono inviati i messaggi. Gli *Exchange* ricevono un messaggio e lo instradano verso *zero* o *più* *code*. L'algoritmo di instradamento utilizzato dipende dal *tipo di Exchange* e dalle regole chiamate *Binding*. Hanno le seguenti proprietà:
  + **Nome**
  + **Durabilità** (può essere durevole o transitorio)
  + **Auto-delete** (l'exchange viene eliminato quando tutte le code hanno terminato di usarlo)
  + **Argomenti** (hash map, dipendente dal broker)
* **Binding (Associazioni)**: Regole che collegano un exchange con un altro exchange o con una coda. Questo è un valore di tipo stringa.
* **Queue (Code)**: Memorizzano i messaggi (in memoria o su disco) fino a quando non vengono consumati dalle applicazioni. Le code hanno le seguenti proprietà:
* **Nome**
* **Durevole** (sopravvive a un riavvio del broker)
* **Esclusiva** (utilizzata da una sola connessione, il che significa che la coda verrà eliminata quando la connessione viene chiusa)
* **Auto-delete** (la coda viene eliminata quando l'ultimo consumatore si disiscrive)
* **Argomenti** (hash map, dipendente dal broker)

Il modello AMQP fornisce quattro tipi di exchange:

* **Default Exchange**: è un direct exchange senza nome (stringa vuota) pre-dichiarato dal broker. Ogni coda che viene creata è automaticamente associata a esso con una routing key che è la stessa del nome della coda. IE, se creo una coda con il nome *search-indexing-online*, il broker AMQP la assocerà al default exchange utilizzando *search-indexing-online* come routing key (in questo contesto talvolta chiamata binding key). Pertanto, un messaggio pubblicato sull'exchange predefinito con la routing key *search-indexing-online* verrà instradato verso la coda *search-indexing-online*.
* **Direct exchange** [amq.direct]: Un direct exchange consegna i messaggi alle code in base alla **routing key** del messaggio. Un direct exchange è ideale per l'instradamento unicast dei messaggi. Possono essere utilizzati anche per l'instradamento multicast. IE:
  + Una coda si associa all'exchange con una routing key **K**
  + Quando un nuovo messaggio con la routing key **R** arriva al direct exchange, l'exchange lo instrada alla coda se **K = R**.
  + Se più code sono collegate a un direct exchange con la stessa routing key **K**, l'exchange instraderà il messaggio a tutte le code per le quali **K = R**.
* **Fanout exchange** [amq.fanout]:Un fanout exchange instrada i messaggi a tutte le code ad esso collegate e la routing key viene ignorata. Se **N** code sono collegate a un fanout exchange, quando un nuovo messaggio viene pubblicato su quell'exchange, una copia del messaggio viene consegnata a tutte le **N** code. I fanout exchanges sono ideali per l'instradamento **broadcast** dei messaggi.
* **Topic exchange** [amq.topic]: Questo exchange è simile al direct exchange, con la differenza che accetta wildcard (espressioni regolari) per le routing key, utilizzando:
  + \* (può sostituire esattamente una parola)
  + # (può sostituire zero o più parole)
* **Headers exchange** [amq.match, amq.header]: Questo exchange instrada i messaggi confrontando più header. È necessario specificare se gli header devono corrispondere esattamente aggiungendo x-match:all (header: key), oppure se deve essere sufficiente una corrispondenza parziale aggiungendo x-match:any (header: key) nell’header del messaggio.
* **Consistent Hash Exchange**: E’ un tipo di exchange specializzato che distribuisce i messaggi tra più code utilizzando un algoritmo di hashing consistente.

Di seguito un diagramma esplicativo:

