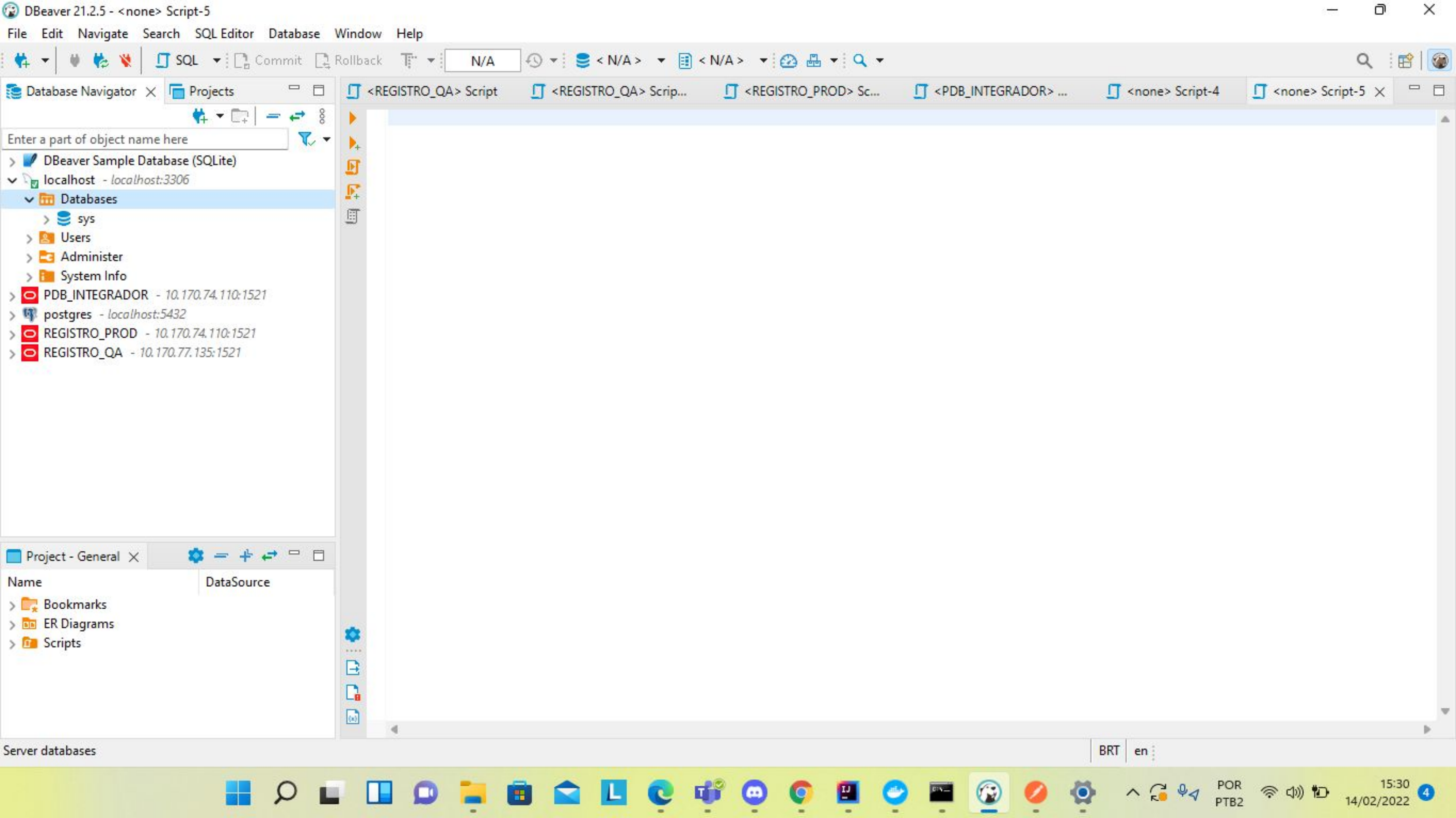
Starting app:

-- First of all, we have a file called docker-compose.yml, it files contains all infrastructure to run our app;

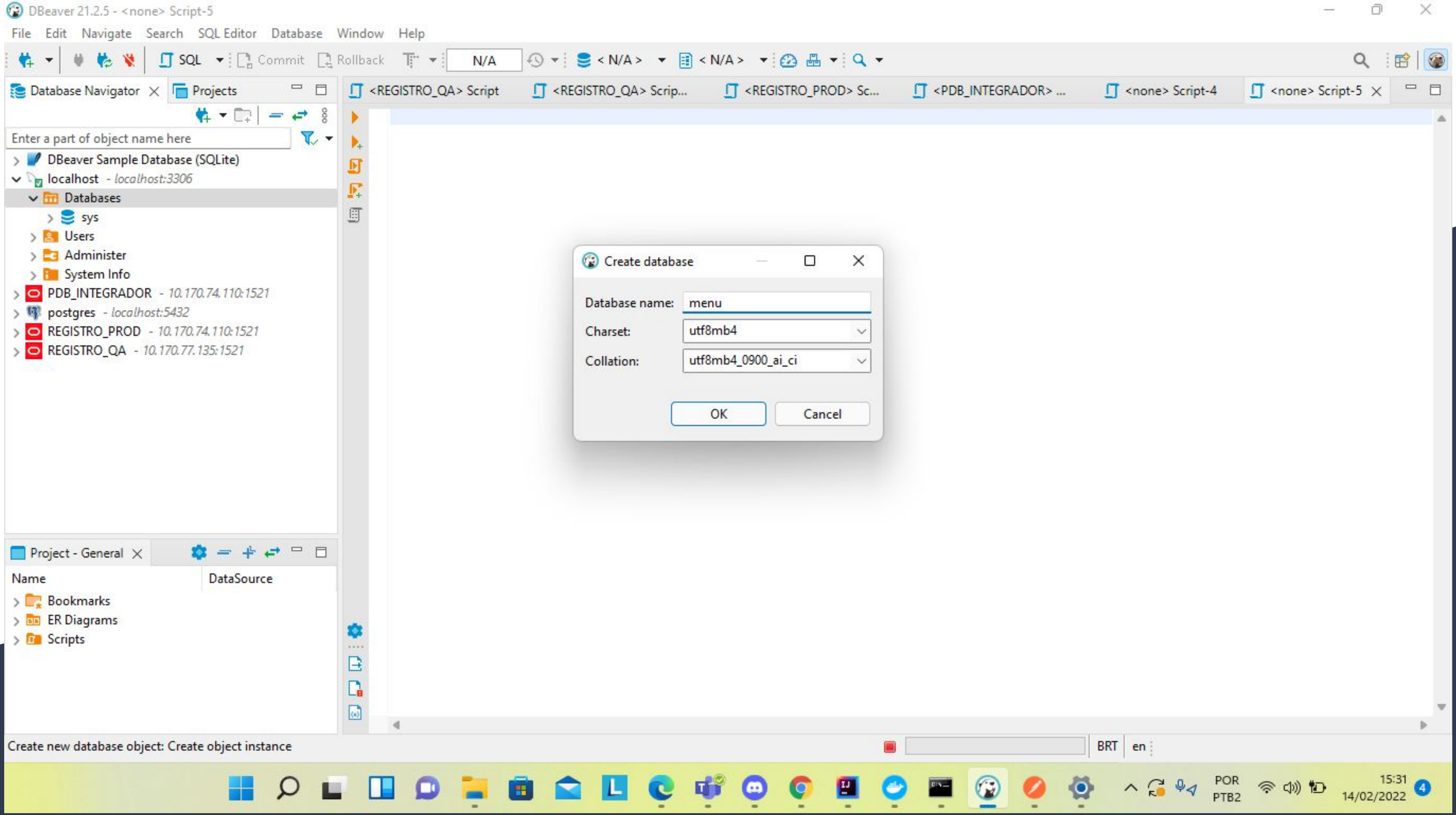
-- to run this file you just run docker-compose up in this file path, if you prefer running in detached way, it is easy, just use -d : docker-compose -d up;

After this first step is necessary to do some configuration in your database and in rabbit MQ, you can to see this configurations in nexts slides.

Configuring data base



It is necessary to create a new database called menu



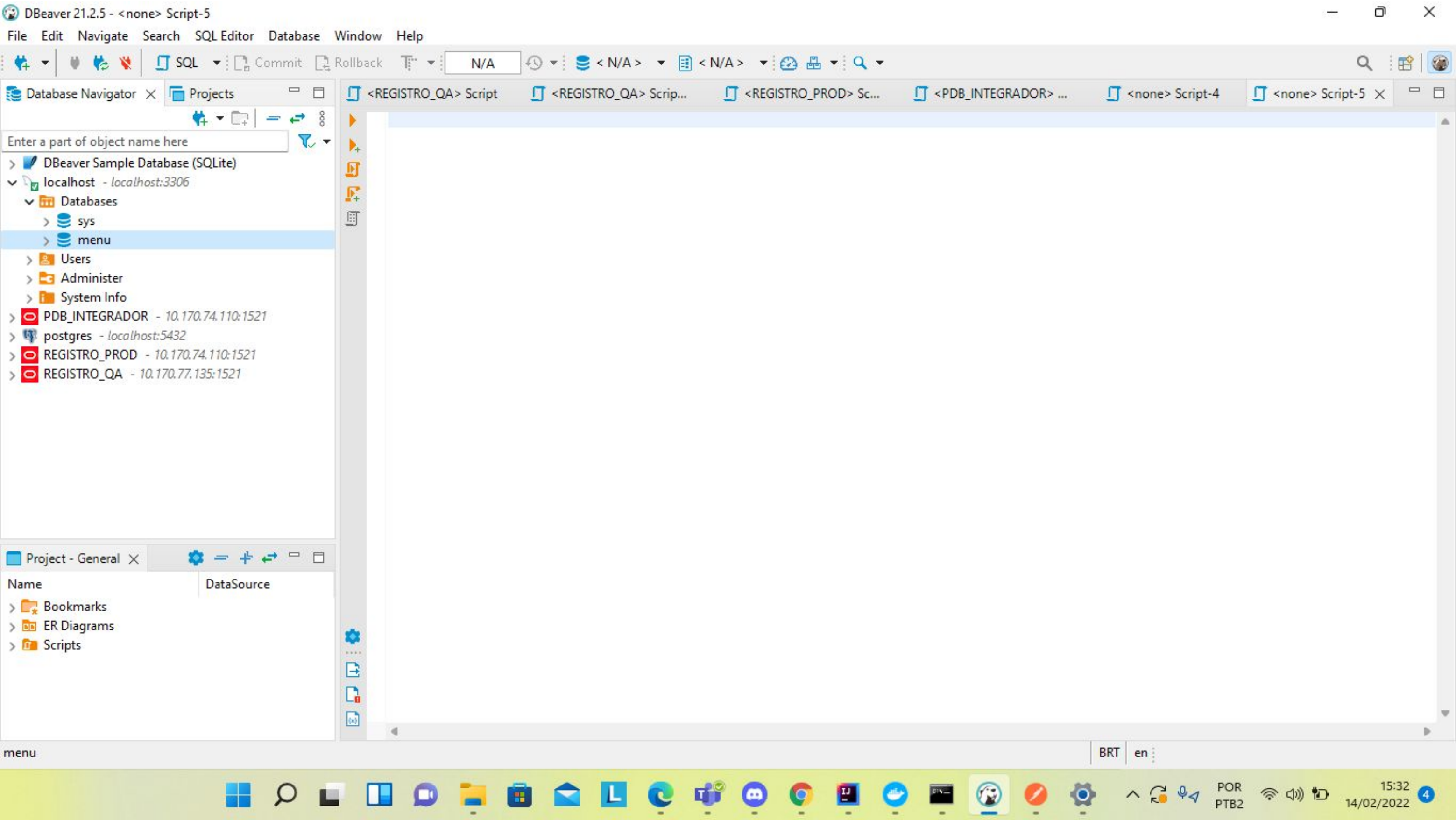
Database name: menu

Charset: utf8mb4

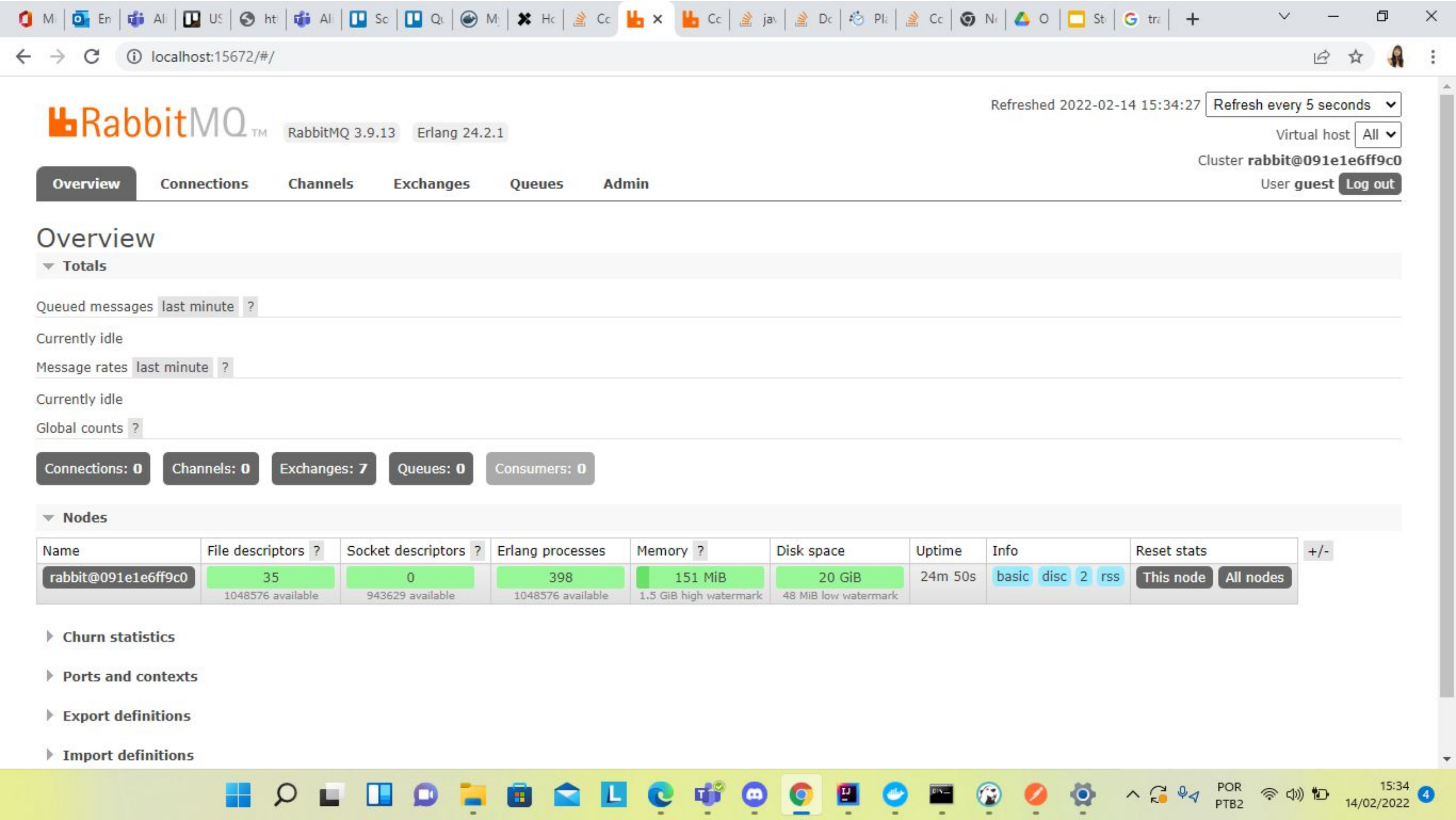
Collation: utf8mb4_0900_ai_ci

Create new database object: Create object instance

It is necessary to create a new database called menu



Configuring rabbit MQ



RabbitMQ 3.9.13 Erlang 24.2.1

Refreshed 2022-02-14 15:34:27 Refresh every 5 seconds

Virtual host All

Cluster rabbit@091e1e6ff9c0

User guest

Overview

Totals

Queued messages last minute ?

Currently idle

Message rates last minute ?

Currently idle

Global counts ?

Connections: 0 Channels: 0 Exchanges: 7 Queues: 0 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats
rabbit@091e1e6ff9c0	35 1048576 available	0 943629 available	398 1048576 available	151 MiB 1.5 GiB high watermark	20 GiB 48 MiB low watermark	24m 50s	basic disc 2 rss	This node All nodes

+/-

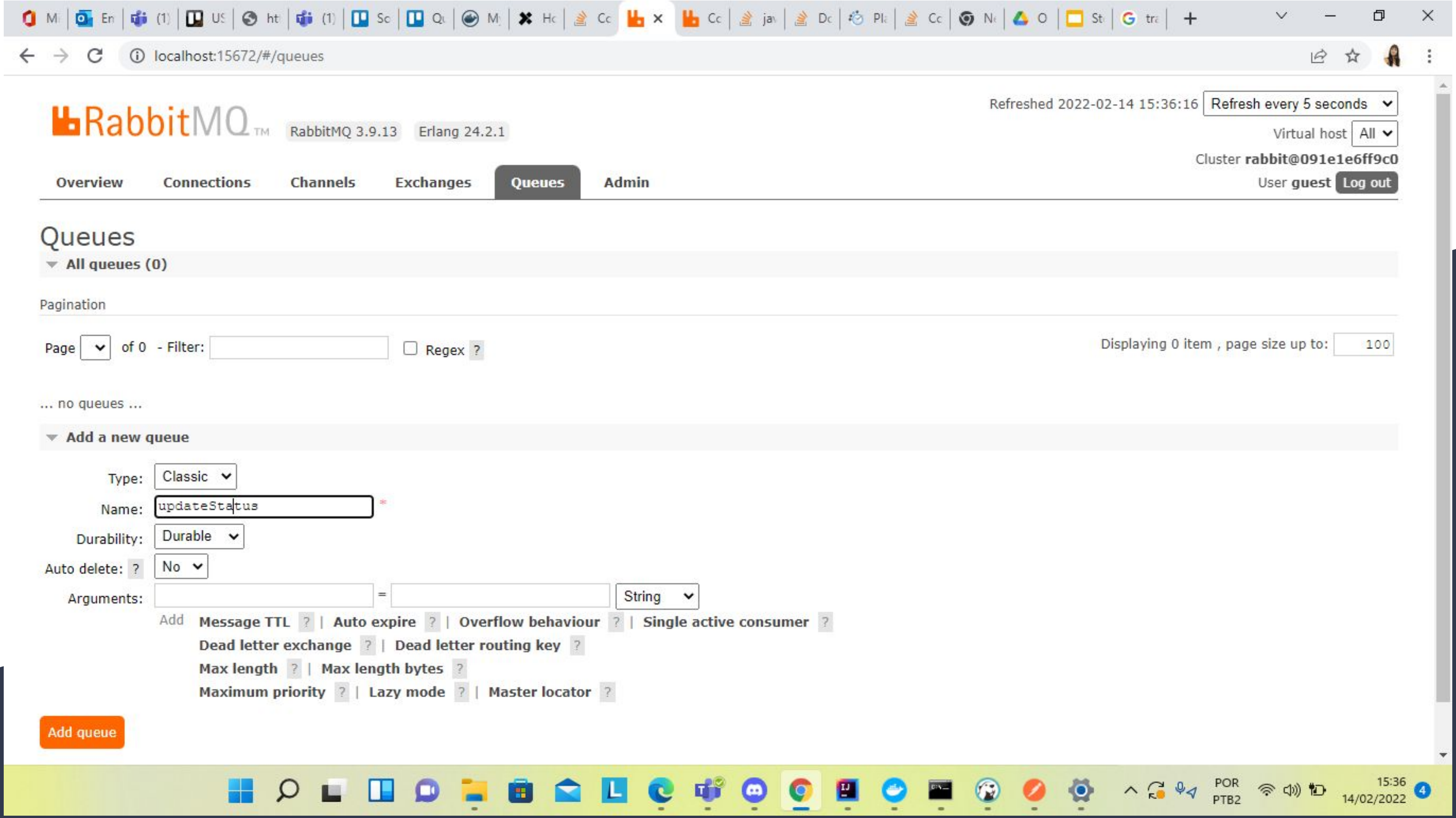
Churn statistics

Ports and contexts

Export definitions

Import definitions

It is necessary to create a new queue with name updateStatus



localhost:15672/#/queues

RabbitMQ 3.9.13 Erlang 24.2.1

Refreshed 2022-02-14 15:36:16 Refresh every 5 seconds

Virtual host All

Cluster rabbit@091e1e6ff9c0

User guest Log out

Queues

All queues (0)

Pagination

Page of 0 - Filter: □ Regex ?

Displaying 0 item , page size up to: 100

... no queues ...

Add a new queue

Type: Classic

Name: updateStatus *

Durability: Durable

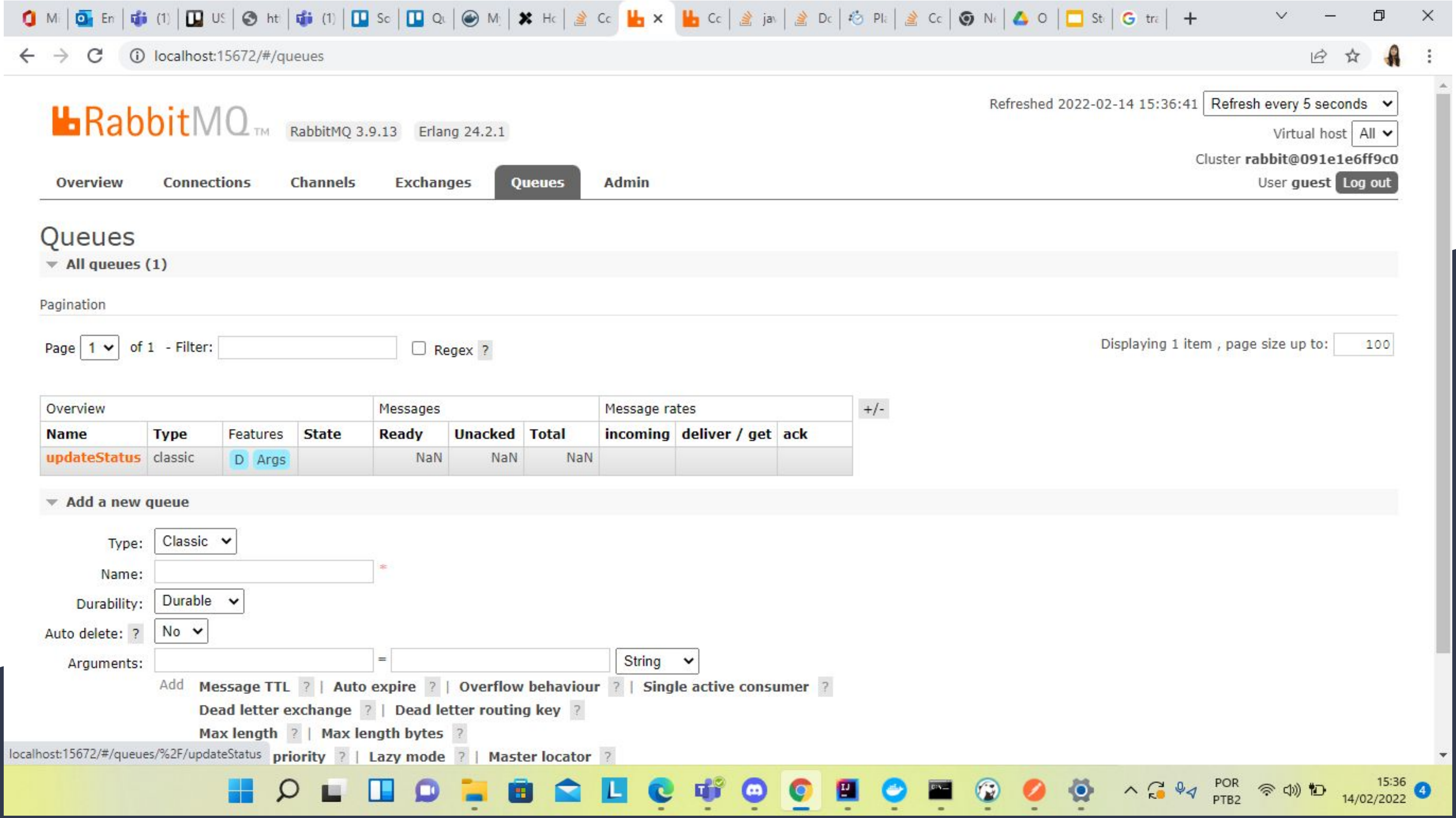
Auto delete: ? No

Arguments: = String

Add Message TTL ? | Auto expire ? | Overflow behaviour ? | Single active consumer ?
Dead letter exchange ? | Dead letter routing key ?
Max length ? | Max length bytes ?
Maximum priority ? | Lazy mode ? | Master locator ?

Add queue

It is necessary to create a new queue with name updateStatus



localhost:15672/#/queues

RabbitMQ 3.9.13 Erlang 24.2.1

Refreshed 2022-02-14 15:36:41 Refresh every 5 seconds

Virtual host All

Cluster rabbit@091e1e6ff9c0

User guest Log out

Queues

All queues (1)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Displaying 1 item , page size up to: 100

Overview				Messages			Message rates		
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
updateStatus	classic	D Args		NaN	NaN	NaN			

Add a new queue

Type: Classic

Name: *

Durability: Durable

Auto delete: ? No

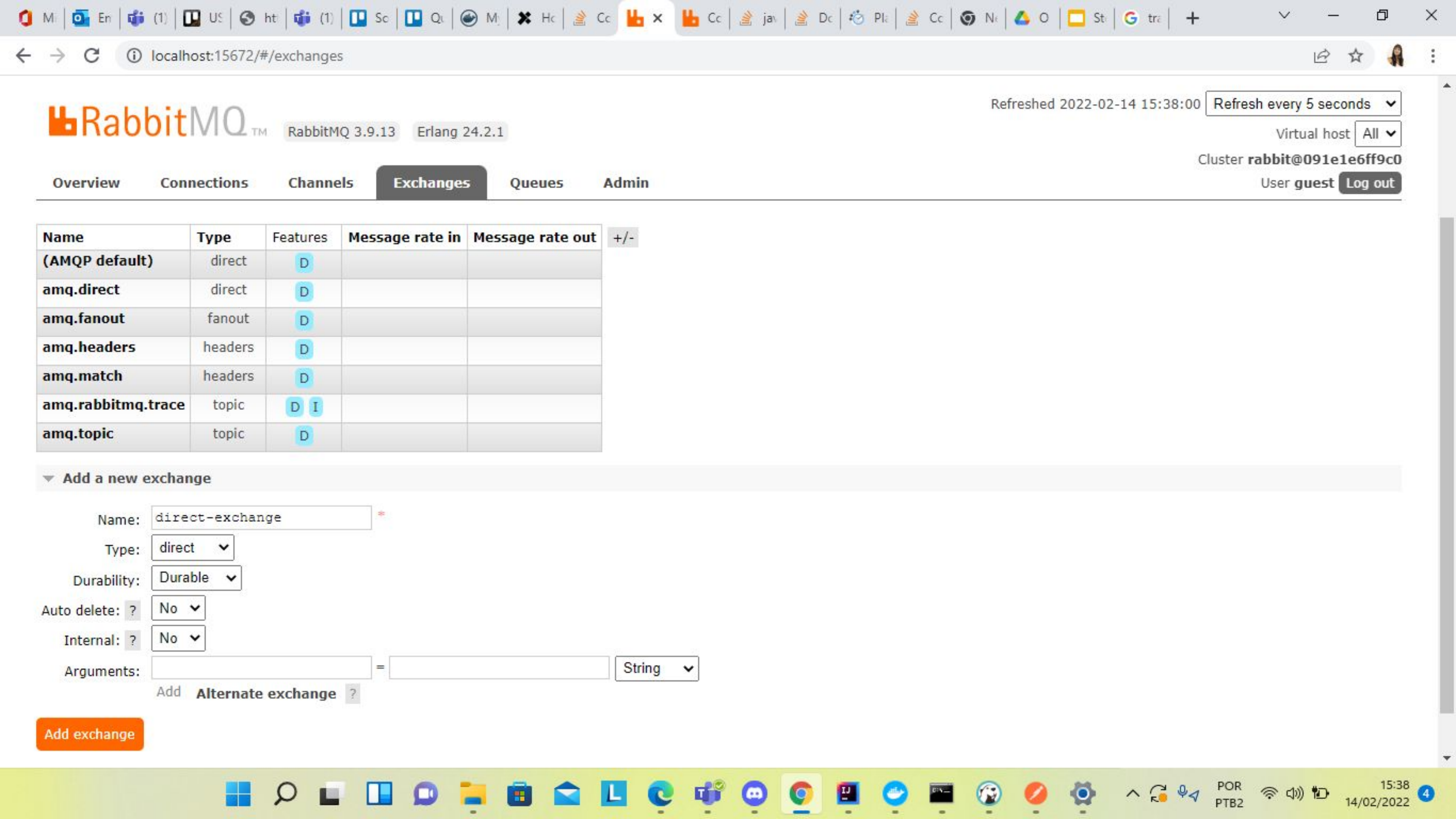
Arguments: = String

Add Message TTL ? | Auto expire ? | Overflow behaviour ? | Single active consumer ?
Dead letter exchange ? | Dead letter routing key ?
Max length ? | Max length bytes ?
priority ? | Lazy mode ? | Master locator ?

localhost:15672/#/queues/%2F/updateStatus

POR PTB2 15:36 14/02/2022

It is necessary to create a new exchange called direct-exchange



localhost:15672/#/exchanges

RabbitMQ 3.9.13 Erlang 24.2.1

Refreshed 2022-02-14 15:38:00 Refresh every 5 seconds

Virtual host All

Cluster rabbit@091e1e6ff9c0

User guest

Name	Type	Features	Message rate in	Message rate out
(AMQP default)	direct	D		
amq.direct	direct	D		
amq.fanout	fanout	D		
amq.headers	headers	D		
amq.match	headers	D		
amq.rabbitmq.trace	topic	D I		
amq.topic	topic	D		

Add a new exchange

Name: direct-exchange *

Type: direct

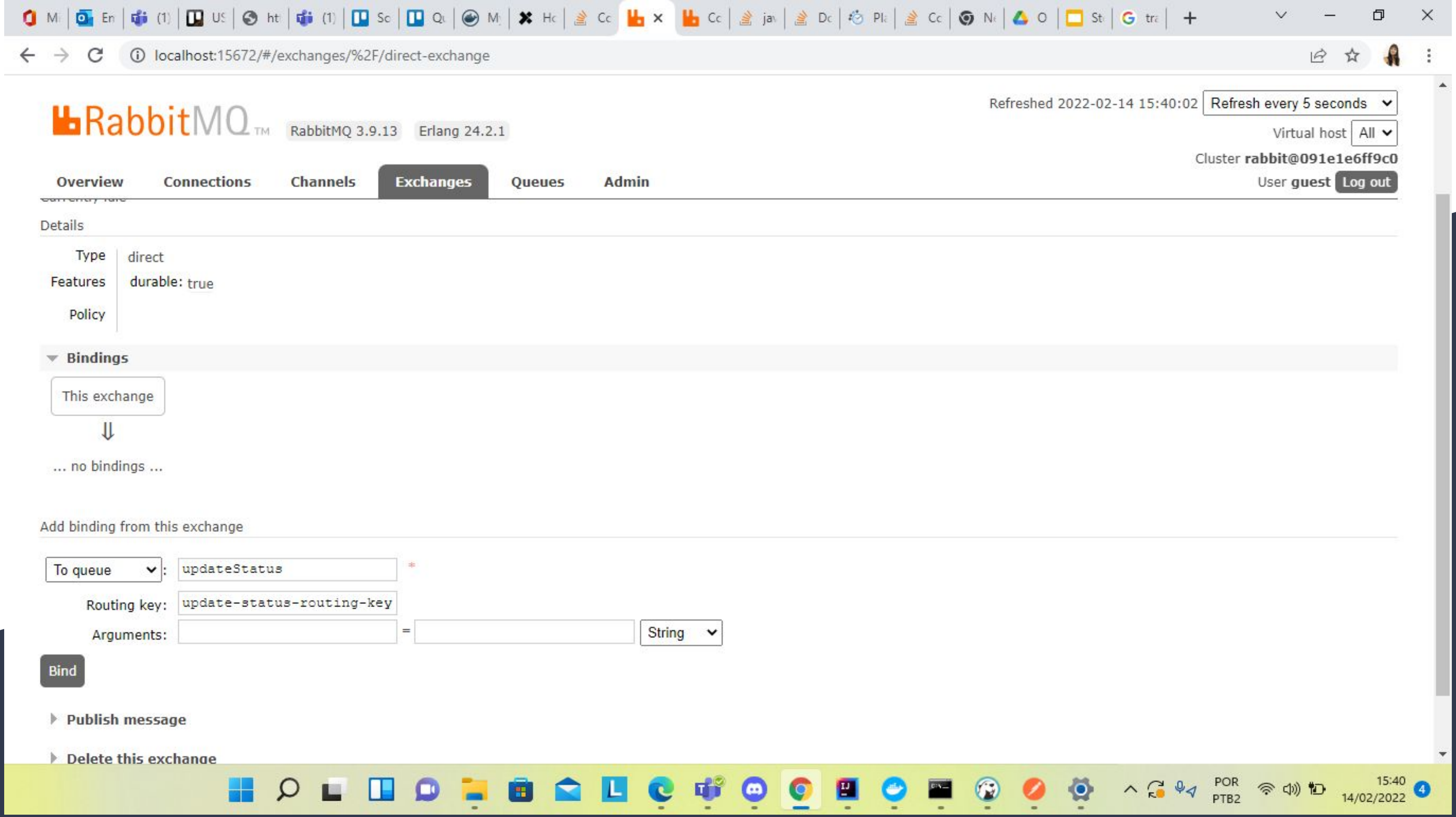
Durability: Durable

Auto delete: ? No

Internal: ? No

Arguments: = String

It is necessary to do a bind between queue and exchange, use update-status-routing-key to routing key and updateStatus to queue



localhost:15672/#/exchanges/%2F/direct-exchange

RabbitMQ 3.9.13 Erlang 24.2.1

Refreshed 2022-02-14 15:40:02 Refresh every 5 seconds

Virtual host: All

Cluster rabbit@091e1e6ff9c0

User guest

Details

Type: direct

Features: durable: true

Policy:

Bindings

This exchange

⇓

... no bindings ...

Add binding from this exchange

To queue: updateStatus *

Routing key: update-status-routing-key

Arguments: = String

Testing app:

- Next step is to start our app;

And you can access on port **8079**

We have two endpoints in this app:

- http://localhost:8079/products:

It is a POST endpoint used to create new products;

http://localhost:8079/products/update-status/2

It is a PUT endpoint used to update product´s status:

- If your products is with ACTIVATE status a request to this endpoint will change products´s status to INACTIVE.

- If your products is with INACTIVE status a request to this endpoint will change products´s status to ACTIVE.

I will send a postman collection with this material.

Thanks for much for this opportunity, the next features will be:

- Modularization;
- Authentication;
- Unit tests;
- Swagger documentation;
- CD/CI with Jenkies;
- Kube files;
- Cloud deploy;