

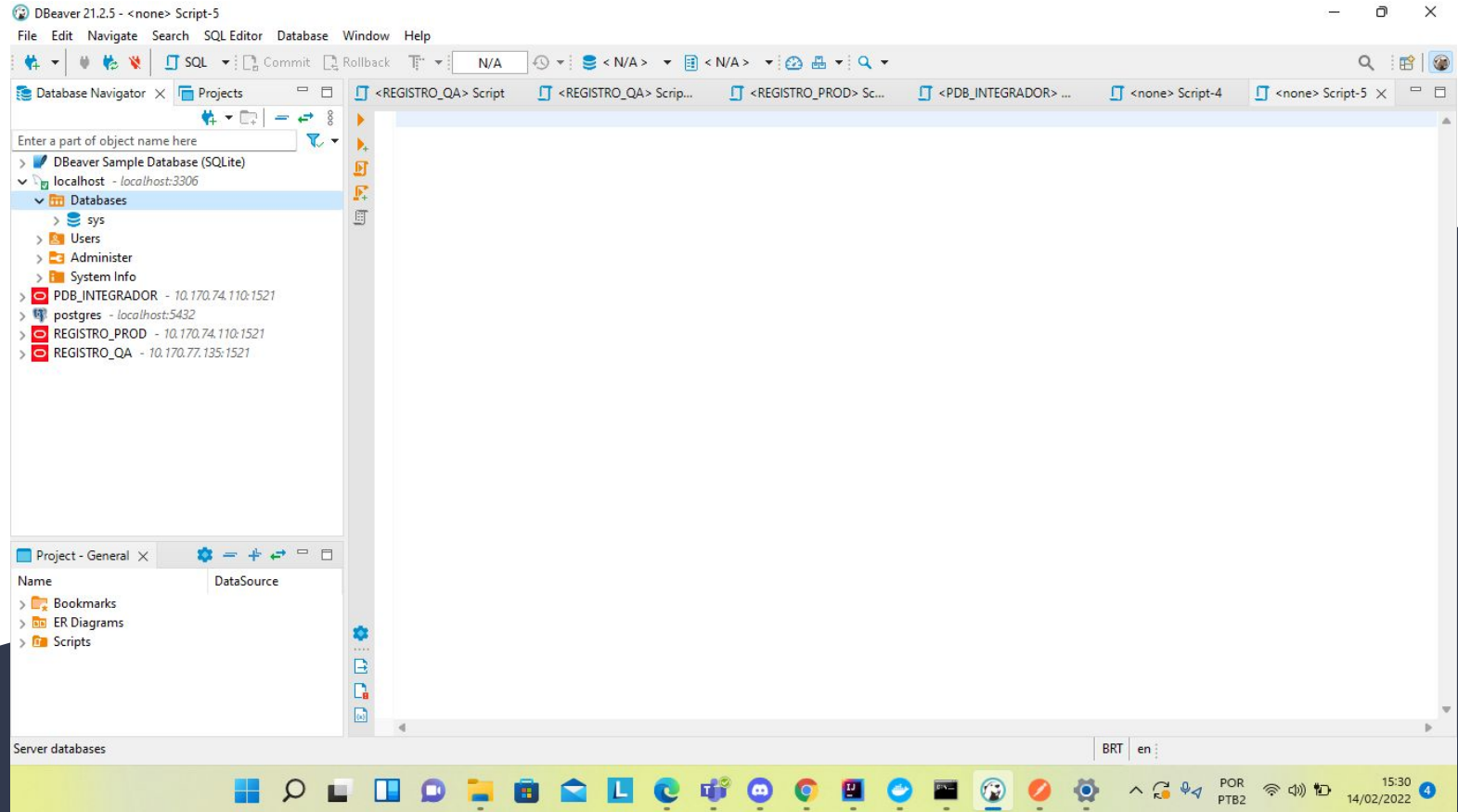
Starting app:

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

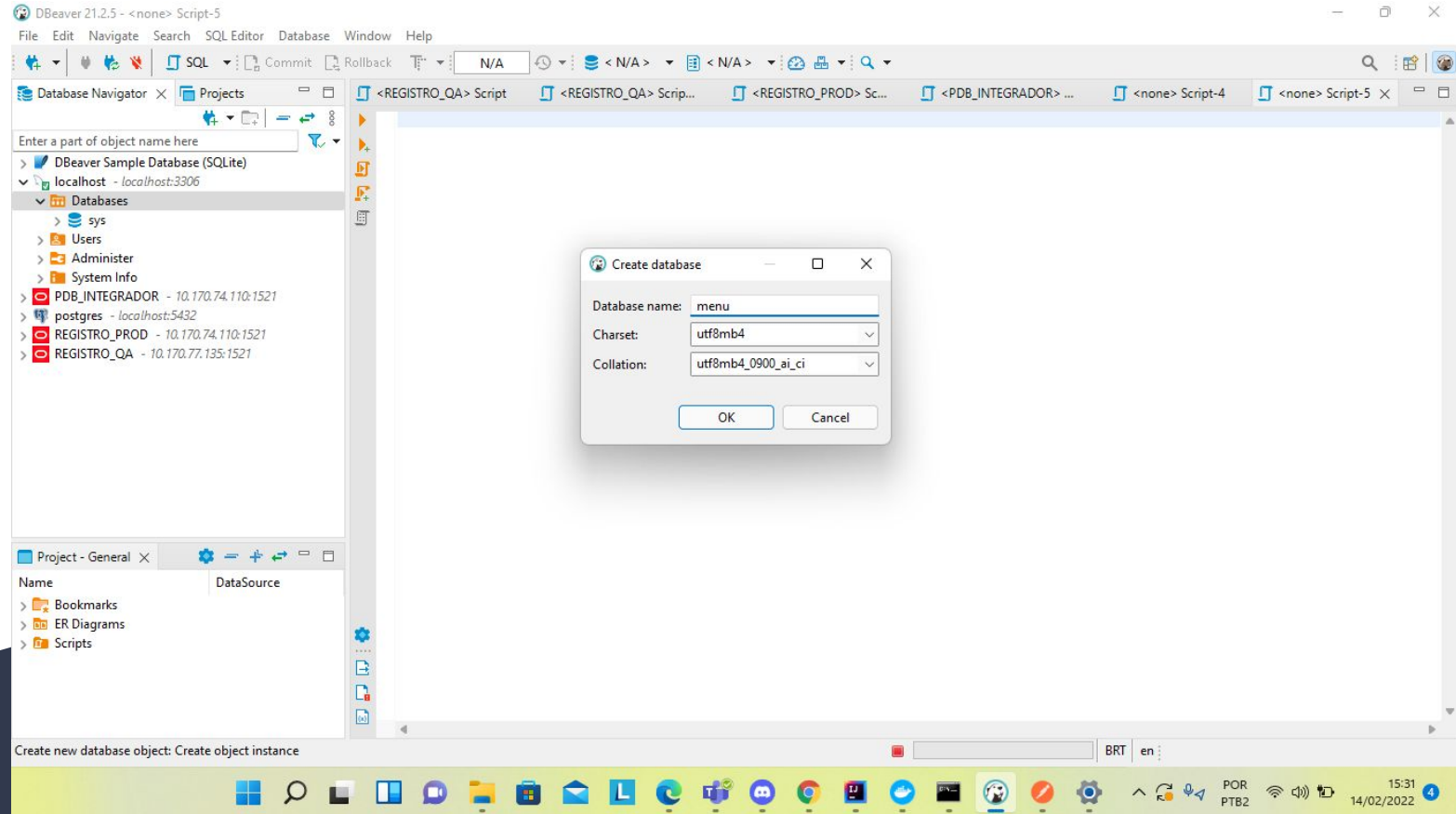
-- to run this file you just run docker-compose up in this file path, if you prefer running in detached way, it is easy, just use -d : docker-compose -d up;

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.

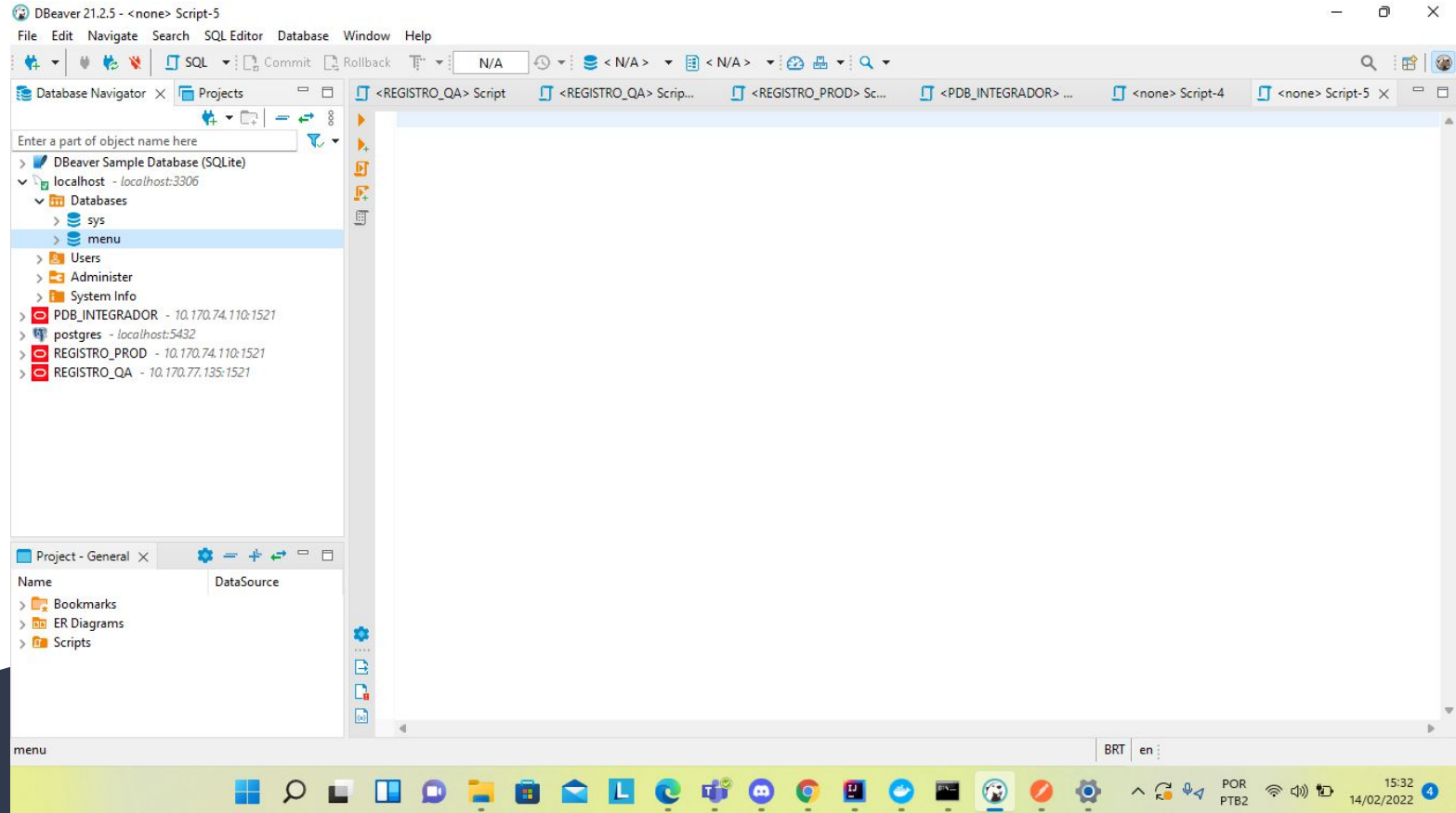
Configuring data base



It is necessary to create a new database called menu



It is necessary to create a new database called menu



Configuring rabbit MQ

The screenshot displays the RabbitMQ management interface in a web browser. The browser's address bar shows the URL `localhost:15672/#/`. The interface header includes the RabbitMQ logo, version information (RabbitMQ 3.9.13, Erlang 24.2.1), and a refresh button set to "Refresh every 5 seconds". The current cluster is identified as `rabbit@091e1e6ff9c0` with the user `guest` logged in. A navigation bar at the top contains tabs for Overview, Connections, Channels, Exchanges, Queues, and Admin.

The main content area is titled "Overview" and features a "Totals" section with the following metrics:

- Queued messages: last minute ?
- Currently idle
- Message rates: last minute ?
- Currently idle
- Global counts ?

Below these metrics, a row of buttons displays the following counts:

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

The "Nodes" section contains a table with the following data:

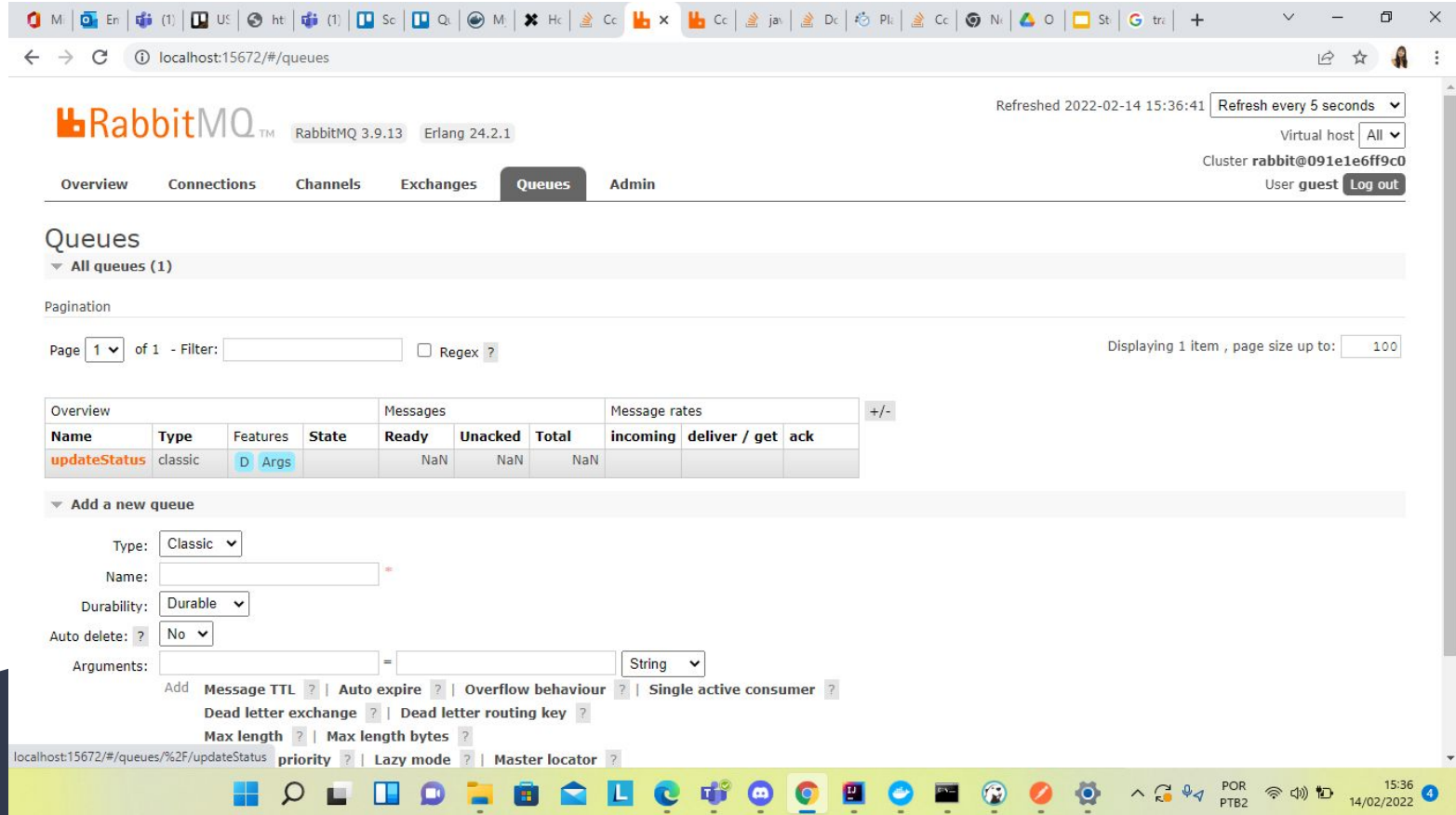
Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@091e1e6ff9c0	35 1048576 available	0 943629 available	398 1048576 available	151 MiB 1.5 GiB high watermark	20 GiB 48 MiB low watermark	24m 50s	basic disc 2 rss	This node All nodes	

At the bottom of the Overview page, there are links to "Churn statistics", "Ports and contexts", "Export definitions", and "Import definitions". The Windows taskbar at the bottom of the screen shows the time as 15:34 on 14/02/2022.

It is necessary to create a new queue with name updateStatus

The screenshot shows the RabbitMQ web management interface in a browser. The address bar indicates the URL is `localhost:15672/#/queues`. The top navigation bar includes tabs for Overview, Connections, Channels, Exchanges, Queues (selected), and Admin. The 'Queues' section is active, showing 'All queues (0)'. Below this, there's a pagination section with 'Page 1 of 0' and a filter input. A message states '... no queues ...'. The 'Add a new queue' section is expanded, showing a form with the following fields: Type (Classic), Name (updateStatus), Durability (Durable), Auto delete (No), and Arguments (empty). Below the arguments field, there are links for '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', and 'Master locator'. An 'Add queue' button is at the bottom left of the form. The top right of the interface shows 'Refreshed 2022-02-14 15:36:16', a 'Refresh every 5 seconds' dropdown, 'Virtual host All', 'Cluster rabbit@091e1e6ff9c0', and 'User guest' with a 'Log out' button. The bottom taskbar shows various application icons and the system clock indicating 15:36 on 14/02/2022.

It is necessary to create a new queue with name `updateStatus`



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)

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	Durable		NaN	NaN	NaN			

▼ Add a new queue

Type: Classic

Name:

Durability: Durable

Auto delete: No

Arguments: = String

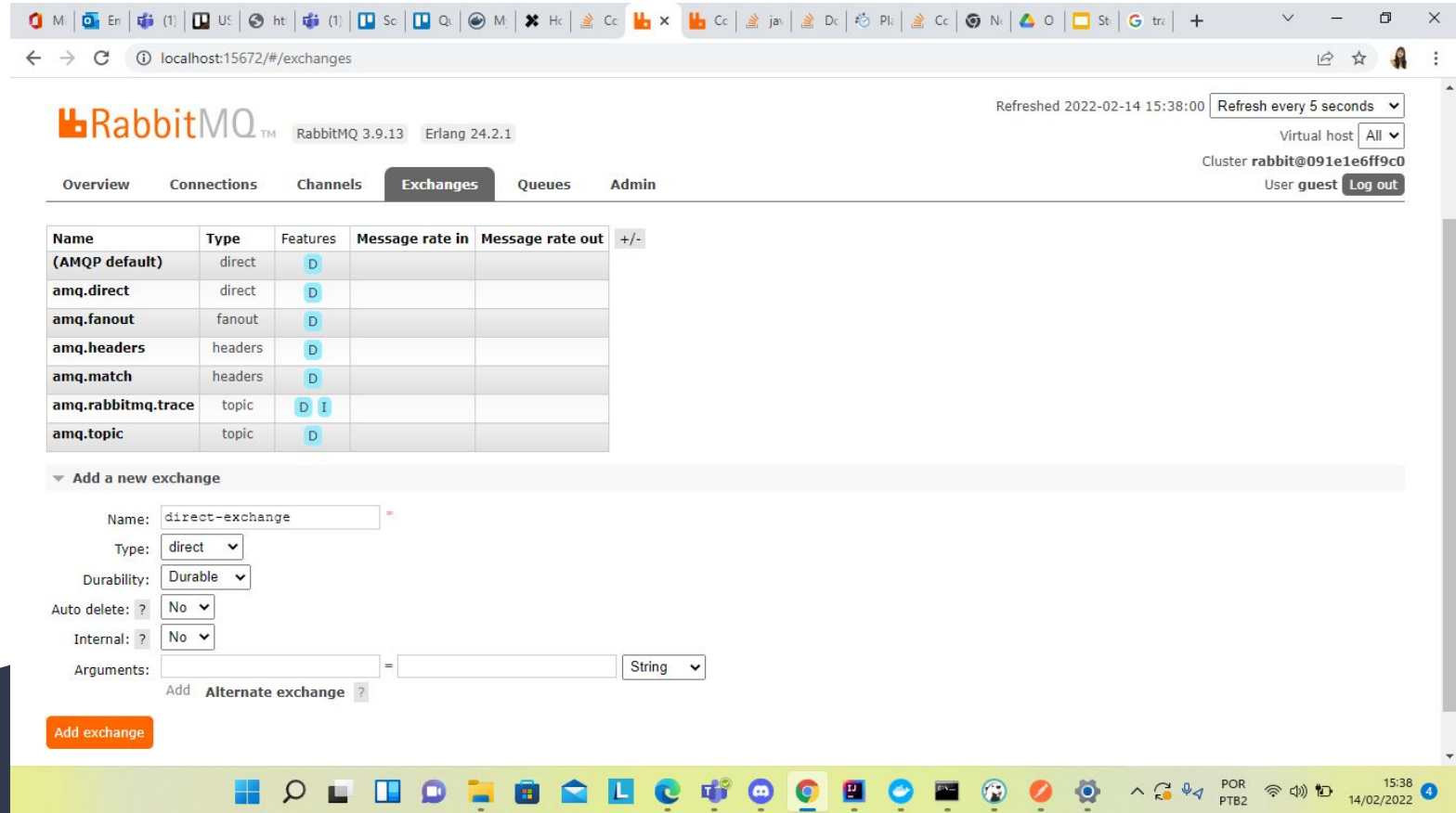
Add Message TTL ? | Auto expire ? | Overflow behaviour ? | Single active consumer ?

Dead letter exchange ? | Dead letter routing key ?

Max length ? | Max length bytes ?

priority ? | Lazy mode ? | Master locator ?

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



The screenshot shows the RabbitMQ Admin interface in a web browser. The browser's address bar shows `localhost:15672/#/exchanges`. The page title is "RabbitMQ" with version information "RabbitMQ 3.9.13" and "Erlang 24.2.1". The page is refreshed every 5 seconds. The "Exchanges" tab is selected in the top navigation bar. The main content area displays a table of existing exchanges and a form to add a new one.

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: *

Type: ▼

Durability: ▼

Auto delete: ? ▼

Internal: ? ▼

Arguments: = ▼

Add ?

The bottom of the screenshot shows a Windows taskbar with various application icons and a system tray on the right displaying the time as 15:38 on 14/02/2022.

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

The screenshot shows the RabbitMQ web interface in a browser window. The address bar indicates the URL is `localhost:15672/#/exchanges/%2Fdirect-exchange`. The interface is for RabbitMQ 3.9.13 running on Erlang 24.2.1. The 'Exchanges' tab is selected, showing details for a 'direct' exchange named 'direct-exchange'. The exchange is durable and has no bindings. Below the 'Bindings' section, there is a form to 'Add binding from this exchange'. The 'To queue' dropdown is set to 'updateStatus'. The 'Routing key' is set to 'update-status-routing-key'. The 'Arguments' field is empty, and the 'String' dropdown is set to 'String'. A 'Bind' button is visible below the form. At the bottom of the interface, there are links to 'Publish message' and 'Delete this 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 Log out

Overview Connections Channels **Exchanges** Queues Admin

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

Bind

► Publish message

► Delete this exchange

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 Jenkins;
- Kube files;
- Cloud deploy;