Msg Exchange Service

Problem statement:

Build an application or set of applications to acheive the following:
1. receive messages from source/sources
2. persist messages in a locally hosted database
3. forward messages to sink/destination.
4. Provide instructions on how to execute your application/s in commandline and or install any toolsuits or frameworks/applications/environment [provided they are opensource/openly available], and how to compile it from source.

Constrains:

get: localhost:8181/testbench/config [application/json or application/xml]
Will return {config:[{source,source_rate}], sink:sink_rate}
this will list sourceNames and max allowed message rate. So {source1: 10} means messages with source = source1 should have a maximum rate of 10 msg/s
and ,sink: 20} means you should forward messages at a max rate of 20 msg/s to the final destination

Source messages:

post: localhost:8182/testsolution/rest/msg [application/json or application/xml]
Solution must allow the above post. For msg with fields

```
    String ref;
    String source;
    String payload;
```

Handle apropriately the case where source has more than source_rate messages in a given second.

Persistance:

Messages should be saved in a table in the connection shown
jdbc:mysql://localhost:3306/msg_exchange
user: exchange
password: 3xc#@n8e
Should you require/use a different connection it should at least be an RDBMS running on localhost with clear download and setup instructions so that your solution can be tested locally.

Add a status field which will be updated to reflect the different stages of processing
each msg must be persisted on initial receipt and when eventually received at sink.
If sink fails update msg status accordingly.
Include at least the following status {RECIEVED_AT_EXCHANGE, RECEIVED_AT_SINK} and any other statuses reflective of the current state/position in the exchange.

Destination:

post: 8181/testbench/sink/msg [application/json or application/xml]
fields expected:

```
    String id;
```

String source;
String payload;
Where id is a globally unique reference that is generated in solution.
The sink will log every msg received and print accepted/ignored based on the sink_rate. Note: the sink does not indicate that it has ignored the msg because of rate limit. The logging is only for testing your solution. On our test environment your application cannot read the sink log.
After 1000ms with no msg received the sink will print total accepted/ignored and total send

Rules:

Solution will be measured on:

1) Repeatebility. It should be possible to setup and run solution.
2) The ability to forward some messages to the destination
3) Ability to persist all accepted messages to the database.
4) Quality of code, best practices, parttens and unit/integration tests.
5) Management of source/sink rate.
6) Solution optimization

Testing and scoring

The testing application is available on gitlab and will run on port 8181 posting source messages to
http://*localhost:8182/testsolution/rest/msg*
https://github.com/EconetWireless/developer-taksk-02