CPSC 313: Distributed and Cloud Computing

Spring 2022

Lab 3: Message based chat

We will be building a distributed chat server for a few labs in a row. As we saw in class, trying to do pure point to point connections is both more difficult and has far more issues than using a messaging server in the middle.

Within the messaging server, there will be the following queues:

- A general chat that will distribute a message to all clients.
  - This channel is named “general”
- A private channel with whatever name you choose that will hold messages only for you

In your program, you will be connecting to RabbitMQ (RMQ) on a local server. That server is:

http://cps-devops.gonzaga.edu – listening on port 5672

The beginning of this process is simply to create an API that exposes the following endpoints/methods:

- Send: accepts a message and a sends it to RMQ
  - http://<url>:<port>/send/?message=<your message>
- Messages: returns the list of messages currently in the queue
  - http://<url>:<port>/messages/
  - This method returns a list of messages:
    - [“test message 1”, “test message 2”, “another message”, “and so on”]

You will be using the “pika” (pip install pika) library to connect with RMQ. For a client-side HTTP library you should probably use either “requests” or “xhttp”. There are others, and you may use whichever library works best for you.

You will create a single python code file named “mess_chat.py” that implements the API. You should also have a python code file named “mess_chat_test.py” that tests your API. Your third file will be “rmq.py” and will contain the implementation for RMQ interaction.

You will have to learn how to use RMQ, though it is very similar in usage to most messaging servers. I recommend you develop a class for handling all RMQ interactions and use that class to implement your APIs. This file should have a test file “rmq_test.py” that tests the RMQ interactions.

- Both unittest and pytest are good test frameworks for python. I tend to use both.
- To run the tests, you can use appropriate commands in your powershell or terminal
- However, I strongly recommend you use VSCode to run your tests. It’s easy to integrate either pytest or unittest (or both) into VSCode to run your tests.
- You should be getting used to using postman (postman.com) to help you manually test your API’s