Manual on How To Run UCI With Zero MQ

1. We need to install a couple of libraries that are required for our program. These libraries are as follows:

```
sudo apt-get update && sudo apt-get install -y \

    uuid-dev \

    libssl-dev \

    libnorm-dev \

    libprotokit-dev \

    libzmq3-dev \

    libsodium-dev \

    libapr1-dev \

    libuuid1 \

    libgps-dev \

    libdbus-1-dev \

    libsystemd-dev \

    libcap-dev \

    libgssapi-krb5-2 \

    libpgm-dev
```

2.They must put the IP to which the program is going to connect in the brokerURI property, they must only change the localhost for the public IP:

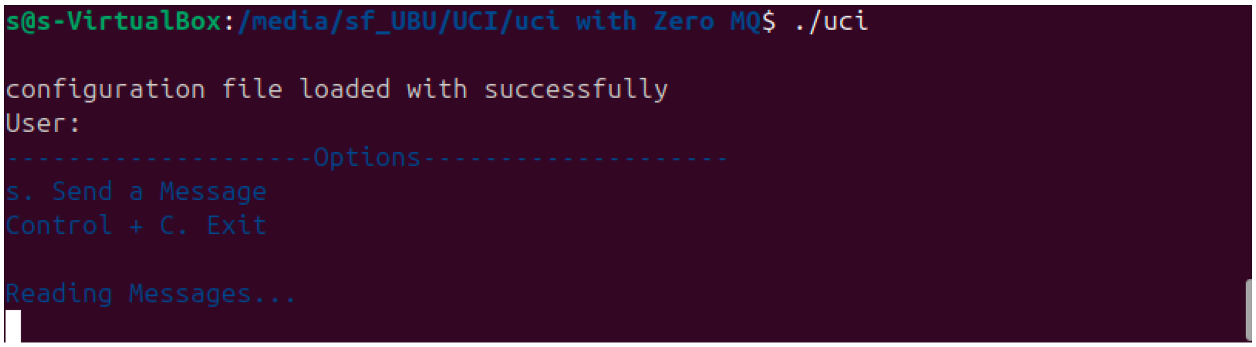
Uci/Conf.xml

There you will find the following properties:

```
<Conf>
   <Properties type="brokerURI">tcp://localhost:61616</Properties>
   <Properties type ="filenameMessage">Messages.xml</Properties>
   <Properties type="key-openssl">a3f9b6c8d2e4f1a7c6e8b9d0a1f2c3e4d5f6a7b8c9d0>
   <Properties type="iv-openssl">4f3b2a1d9c8e7f6d5b4a3c2f1e0d9a8b</Properties>
</Conf>
```

3.Finally, we can already run the program, we enter uci/ and execute:
./uci

If everything goes well, our execution should look like this:



```
s@s-VirtualBox:/media/sf_UBU/UCI/uci with Zero MQ$ ./uci

configuration file loaded with successfully
User:
--------------------Options--------------------
s. Send a Message
Control + C. Exit

Reading Messages...
```

In the first instance our program is in read mode, that is, it is listening to any message that is sent on the ip and port in which we are connected, these messages are sent encrypted and when receiving them decrypted, when receiving the messages we receive the following parameters:

```
Received message: hola como están?

with uuid:ba44f943-506a-4091-aa62-e96ba3b8032c

date Time: 2024-08-26 11:35:21

Latitude and Longitude:  0.000000,0.000000
```

If we want to send a message we press the letter : s on our keyboard and then we enter the message we want to send and we give enter, the program will again enter in read mode, if we want to exit the program we press the letter : Control + C.

All messages sent and received are stored at the root of where our program is in a file called: Messages.xml

There is the UUID, date time, latitude and longitude, and of course our message.

Note:

In this version of the program with Zero MQ the functionality regarding the sending and receiving of messages continues to work in the same way in the same way its workflow however there are a couple of features that in this version of the program with Zero MQ are not, below I will list them:

1) The authentication system that we have configured in the version of the program with Active MQ here is not here because Zero MQ does not have a basis to implement it

2) The durable message system that we have in the Active MQ version is not here, this is because as we do not have a centralized server we do not have a way to make our messages persist.

3) In this version of Zero MQ the connection channels with connected user identification are not connected either, because the way Zero MQ works bidirectional is not like in Active MQ that we have a centralized server to which we configure certain channels with users and identifiers