



SOFE 4790U: Distributed Systems (Fall 2023)

Instructor: Dr. Ahmed Badr

Adapted from content developed by Dr. Q.H. Mahmoud

Lab#3 CRN 43525 – Developing Distributed Applications with ZeroMQ

Objective

The objective of this lab is to get started with building distributed applications using a message-oriented middleware called ZeroMQ. You will modify and extend the sample programs attached with this lab (*ZClient.java*, *ZServer.java*). You will modify and extend sample code, and document your experiments and results of running programs in your lab report. You also need to submit the source code for each task, so you may want to create separate folders as you go along for the source code for each task.

Lab time: Thursday, Oct 26 between 02:10pm – 05:00pm

Tasks

Task #1: Establish the connection (20 marks)

Use the provided *ZServer.java* & *ZClient.java* program (available on Canvas). Modify the code in both client and server so that when the client connects to the server, both print a successful message upon start.

See sample code here:

- <https://zeromq.org/languages/java/>
- <https://zeromq.org/get-started/?language=java&library=jeromq#>

Task #2: Prime numbers (20 marks)

In this task, you will get a better understanding of how a client can send a message and get reply from the server. Remember ZeroMQ doesn't know anything about the data you send except its size in bytes. That means you are responsible for formatting it safely so that applications can read it back. Modify the code from task#1 or the *hwserver.java* & *hwclient.java* (available on Canvas) so that client can send an integer number to the server. Server will receive the number and send all the prime numbers up to the provided number. Client receives the message and prints it. For example: If client sends “10”, it will receive “2,3,5,7” from the server.

Task #3: Repeated inputs from client (20 marks)

In this task, the client will send some messages repeatedly to the server and server will reply accordingly. Modify the code so that client can take a string as input from the keyboard and send

it to server. Server will print the string, reverse it and send back to client. The client will receive and print the message received (reversed string), and send another message to the server (another string to be reversed) and the process is repeated until the client sends the “close” message to the server which will stop the connection between server and client will terminate.

Task#4: Publish-subscribe application (40 marks)

Review, compile and run the sample code: *wuserver* and *wuclient* (available on Canvas). Your task is to adapt this code to accomplish the following task: develop a publish-subscribe application in ZeroMQ for population updates according to postal codes. A set of clients will subscribe to the server with different postal codes taken as input from the keyboard. The server will continuously broadcast updates of population of different areas. When the client listens to the stream of updates of the postal code it has specified, it will print the information received.

How does the server determine the postal code and the population of that area?

Tip: Generate at least 10 random postal codes and population numbers on the server (for simplicity declare the postal code as a 4-digit integer number. For example: 1001, 4349, etc.). Let the server prints the postal codes generated, and use them to subscribe at least 3 clients.

Submission Guidelines

Your submission is due by **11:59AM (just before Noon time NOT midnight)** on the next day of your lab session i.e., on next Friday. **No late submissions will be accepted**. Submissions must be through the link provided on Canvas. **No email submissions will be accepted even if they are emailed before the deadline**.

Your submission must consist of a single zipped file (only *.zip is accepted) containing:

- 1) Your source code for each task. Create a folder with a separate subfolder for each task.
 - Source Code
 - Task1
 - Task2
 - Task3
 - ...etc