

## LAB-#2 -ASYNCHRONOUS MESSAGE SERVER

NAME:- REDDAMMAGARI SREE SOUMYA

STUDENT ID -1001646494

### Requirements for running the assignment:

jdk 1.8.0\_191 or above, IDE(Optional)(eclipse/NetBeans)

### Running the assignment and the flow of GUI:

1. Unzip the Reddammagari\_sxr6494.zip file
2. The folder Reddammagari\_sxr6494 inside has three files MessagingServer.java, MessagingClient.java, Message.java
3. If running on terminal cd to the folder and use the following commands:  
(First) javac MessagingServer.java(For Compilation); java MessagingServer(For Running)  
(Second) javac MessagingClient.java(For Compilation); java MessagingClient(For Running)
4. If running on IDE import the project then compile and run MessagingServer.java, MessagingClient.java
5. After running java MessagingServer GUI pops up then the server starts at "localhost:8080"
6. Server displays a log of usernames and currently connected clients
7. Stop server button is available for stopping the server and exiting out.
8. After running java MessagingClient.java GUI pops up asking users the server address to connect to(i.e enter localhost) and a distinct username from those to online users. If the same username is entered it prompts a dialog box saying "User Online" and username cannot be empty.
9. Once a user registers with a server a dialog box pops up saying Welcome username and a log of usernames used since the server started.
10. User can select the check boxes to his left to select active usernames he wants to send messages. He has an option of sending messages to one, subset or all usernames depending upon his selection of checkboxes.
11. To send a message the user has to click on sendmessage button and type message(drag the UI to right to see sendmessage button)
12. Once the user makes his choice he can enter the message in the text field and click on the send button to add messages into the queue of the recipients. Whenever a message is added onto the queue it is stored in the file messages.dat to make it persistent (drag the UI to right to see send button)
13. When the user clicks on check message button the messages stored in the queue are displayed all together and if the queue is empty it displays no messages in the queue. Whenever a message is removed from the queue it is stored in the file messages.dat to make it persistent (drag the UI to right to see checkmessage button)
14. If the server shuts down suddenly or disconnects and restarts back it reads from the messages.dat file to update clients queue and can display messages sent before server crashing.
15. User can exit and disconnect using the disconnect button. User will be removed from active clients/online clients.

## Assumptions:

1. There is no other process running or blocking port 8080
2. Can handle any number of clients at a time
3. Server remains on indefinitely until disconnected
4. Users cannot send a message to unintended recipients.
5. Messages are transferred between users without losing them
6. messages.dat file is not overwritten by any other program
7. messages.dat file does not get corrupted
8. messages.dat file is not deleted unintentionally

## Known Bugs/Limitation:

The program does not allow the case where the user tries to send a message to disconnected recipient or unintended recipient and receive an error message as the program restricts the user to choose only between online clients.

List of messages are immediately displayed and removed from the queue. There isn't a way implemented in the program to keep a backup of older messages displayed if the user wants to lookback.

## References:

JPanel :

<https://www.geeksforgeeks.org/java-swing-jpanel-examples/>

Remove key events:

<https://stackoverflow.com/questions/4472530/disabling-space-bar-triggering-click-for-jbutton>

Setting the state of checkbox:

<http://www.java2s.com/Code/JavaAPI/javax.swing/JCheckBoxisSelected.htm>

Timestamp Setup:

<https://tecadmin.net/get-current-timestamp-in-java/>

Reading and Writing objects into file:

[https://www.spigotmc.org/threads/save-load-hashmap-in-dat-file.132177/?\\_cf\\_chl\\_jschl\\_tk\\_\\_=e437e361783caa93a8a772de9fe4111d0c6d90fc-1585621687-0-AbssqmZb7YmrC7T-y2wXj4VQBvr69HzAhYaUcTPb777lHRm-uxQsSxZbrV09GqjVyu5v8BIQLX20TIKsfl3UjC0aPhUB0iSr66ZIJCSo5qd54TeUcHS3milwyZhKqe426efPXy5F9n6mTgsWj\\_ejoK\\_-OoFm2sXNqxpVznDKLxDRHSaPANDXvQTp3iwxHb5tEVAYqhE-iKaF3NqmXZC3oN0ABKxhpFe83RxtEVddVLVENa4Er7ilu-ZzKH0fYsSkRZrXYU8g8ja0GeGvIS1\\_TFKMa\\_UfgkUnNZd8TnBYWIIbvGX7z6cIKg9eRb\\_Ie5CrTmmKYRp5cCRIN\\_7XK7PCgM](https://www.spigotmc.org/threads/save-load-hashmap-in-dat-file.132177/?_cf_chl_jschl_tk__=e437e361783caa93a8a772de9fe4111d0c6d90fc-1585621687-0-AbssqmZb7YmrC7T-y2wXj4VQBvr69HzAhYaUcTPb777lHRm-uxQsSxZbrV09GqjVyu5v8BIQLX20TIKsfl3UjC0aPhUB0iSr66ZIJCSo5qd54TeUcHS3milwyZhKqe426efPXy5F9n6mTgsWj_ejoK_-OoFm2sXNqxpVznDKLxDRHSaPANDXvQTp3iwxHb5tEVAYqhE-iKaF3NqmXZC3oN0ABKxhpFe83RxtEVddVLVENa4Er7ilu-ZzKH0fYsSkRZrXYU8g8ja0GeGvIS1_TFKMa_UfgkUnNZd8TnBYWIIbvGX7z6cIKg9eRb_Ie5CrTmmKYRp5cCRIN_7XK7PCgM)

General References:

Multi-Threaded Chat Application

<https://www.geeksforgeeks.org/multi-threaded-chat-application-set-1/>

<https://www.geeksforgeeks.org/multi-threaded-chat-application-set-2/>

Queue Creation:

<https://www.geeksforgeeks.org/queue-remove-method-in-java/>