A Simple Chat App In Java実験報告書

2021013 王巍錚

Compiled in Java 1.8

OS: Ubuntu 20.04

Compile command: “javac –release 8 Classname.java”

This is a User Guide on “A Simple Chat App In Java”. It does not contain much information about implementation. Please read the comments in the source code files to know more about how this system is built.

This app achieves three fundamental functions of a chat app(or a SNS app):

1.Sending a message to an online user

2.Sending a message to all the online users

3.Sending a file to an online user

The basic idea behind this app is to use multithreads for users’ sockets so that they can run at the same time and keep waiting for messages from the server. The sever side keeps listening at port 5000. To simplify the system, this app uses a HashMap to store valid users which means there is no real database system.

There are three users who can be used to log in to the system.

ID: Alice Password:111111

ID: Bob Password: 222222

ID: Cindy Password: 333333

To use this app, open a terminal window and go to the server’s directory. Then run the command “java frame/Frame” to start the server. If we see “Listening at 5000” the server is working.

With the server on, please open a few windows for different users. Go to the client’s directory and run the command “java view/View”. After you enter the system, please follow the instructions, input any user’s id and password to log in as the user. If you successfully enter the “Menu” we are ready to move on to test different functions.

1. Display Online Users

When a client sends a message with a type corresponding to “request all online users”, the server would iterate all client threads and return the list of online users’ names as a String object back to the client. Please log in multiple clients to test this function.

2.Group Message:

When a client sends a message with a type corresponding to “send a message to all”, the server will iterate all online users except the sender himself/herself and send the message to them through ObjectOutputStream so that all other clients can receive the message.

To send a group message, please log in all three clients and select “2” from one of the clients. Input what you want to say to others, then you will see all users but the sender client received the message.

3.Direct Message:

When a client sends a message with a type corresponding to “send a direct message”, the server will get the receiver from the message and send the message to the receiver through ObjectOutputStream so that the receiver client can receive the message.

To send a direct message, please log in at two clients. Input the user id of whom you want to send message to and what you want to say, then the receiver will receive the message.

4.Send a File

When a client sends a message with a type corresponding to “send a file”, the server will get the receiver and from the message send the message to the receiver through FileOutputStream so that the receiver client can receive the file under his/her directory.

You can put the file you want to send under the directory of a client and send the file to another client by inputting their username and file name. Then you should be able to find the file under the receiver’s directory

If you do not want to send your own file, you can log in as Alice and send “test.s” with text content “File Transfer Test” to Bob/Cindy. Then you should find “test.s” in Bob/Cindy’s directory.