

Project1-1 Report

1. README

Project consists of 2 java files named as Server and Client with java extension. These files implement the chat application's client and server. To run the application either compile using *javac <filename>* command. General run time of the application is as follows:

- 1- Run the Server application using default port 9998 just typing *java Server* or run on a desired port by typing *java Server <port-number> <timeout>*.
- 2- Run the Client applications withing same port as the server using same syntax *java Client <port-number> <server-IP>*. Note to run on same computer server-IP needs to be localhost.
- 3- Send a message from Client
- 4- Send a message from Server
- 5- Repeat 3 and 4. To exit, type exit command from either of the applications.

2* If any client does run in <timeout> seconds server times out, with a descriptive message.

3* If client does not sends a message for <timeout> seconds, server timeouts and informs the client by sending timeout command. If client tries to send a message at this point it gracefully exits with information about server's timeout.

2. Overview of the program

- a. The advertised port by the server being provided as an argument at execution

To run the server user enters the port number and duration of timeout as an argument. In contrast in the server side user must enter port number and IP address of the server. Following screenshots show this behavior of the chat application.

```
52$ java Server 8888 10
This is the server
Listening to port: 8888
```

```
52$ java Client 8888 localhost
This is the client
Binding to port localhost: 8888
```

- b. Both sides showing the socket address of the other side once the connection has been established.

In the following shots port number and IP addresses for the sockets can be seen. To implement this behavior *getRemoteSocketAddress* method is used.

```
serhat@ThinkPad:~/Dev/networking-projects/Project1-1-serhat-
52$ java Server 8888 10
This is the server
Listening to port: 8888
Client accepted at remote address /127.0.0.1:35382
```

```
serhat@ThinkPad:~/Dev/networking-projects/Project1-1-serhat-cos
52$ java Client 8888 localhost
This is the client
Binding to port localhost: 8888
Connected to server at remote address localhost/127.0.0.1:8888
```

- c. The timeout settings of the connected socket for both sides

In these screenshots you can see the timeout settings for sockets from client side and server side. In the server-side timeout mechanism is implemented for both server-socket and client-socket using following code:

```
serverSocket = new ServerSocket(port);  
serverSocket.setSoTimeout(timeout*1000);  
clientSocket = serverSocket.accept();  
clientSocket.setSoTimeout(timeout*1000);
```

However in the client side we did not specify timeout. Hence in the client side we observe timeout duration as 0 meaning infinity. In contrast in the server side server-socket and client-socket both has 10000 milliseconds timeout duration's.

```
serhat@ThinkPad:~/Dev/networking-projects/Project1-1-serhat-cosk  
52$ java Server 8888 10  
This is the server  
Listening to port: 8888  
Client accepted at remote address /127.0.0.1:35382  
Timeout settings for client response taking time: 10000  
Timeout settings for server not accepting any client: 10000  
Client did not send a message for 10 seconds.  
Timeout!
```

```
52$  
serhat@ThinkPad:~/Dev/networking-projects/Project1-1-serhat-coskun-641  
52$ java Client 8888 localhost  
This is the client  
Binding to port localhost: 8888  
Connected to server at remote address localhost/127.0.0.1:8888  
Timeout settings for client time: 0  
>>> █
```

d. Chat between the server and client.

The chat feature is implemented in a loop format. In a while loop server and client sends each other messages. However, this implementation is limited as server cannot send a message without getting a message from client and vice versa.

```
serhat@ThinkPad:~/Dev/networking-projects/Project1-1-serhat-  
52$ java Server 8888 10  
This is the server  
Listening to port: 8888  
Client accepted at remote address /127.0.0.1:35392  
Timeout settings for client response taking time: 10000  
Timeout settings for server not accepting any client: 10000  
<<< Hey  
>>> Hey  
<<< I am Serhat  
>>> Me too wow  
52$ java Client 8888 localhost  
This is the client  
Binding to port localhost: 8888  
Connected to server at remote address localhost/127.0.0.1:8888  
Timeout settings for client time: 0  
>>> Hey  
<<< Hey  
>>> I am Serhat  
<<< Me too wow
```

e. The keyword being used, and chat being ended.

“exit” is the exiting keyword. In the following shots client sends the exit command and both sides gracefully exit.

```
52$ java Server 8888 10
This is the server
Listening to port: 8888
Client accepted at remote address /127.0.0.1:35398
Timeout settings for client response taking time: 10000
Timeout settings for server not accepting any client: 10000
<<< hey
>>> hey
<<< exit
Exiting the chat application!
```

```
52$ java Client 8888 localhost
This is the client
Binding to port localhost: 8888
Connected to server at remote address localhost/127.0.0.1:8888
Timeout settings for client time: 0
>>> hey
<<< hey
>>> exit
Exiting the chat application!
```

- f. The server closes the connection in response to a timeout.

In the following shots server times out because of the idleness. Client side did not send a message for more than 10 seconds. In response server exits and informs the client by sending timeout command. In response to timeout command client also closes.

```
serhat@ThinkPad: ~/Dev/networking-projects/Project1-1-serhat-coskun-64152$ java Server 8888 10
This is the server
Listening to port: 8888
Client accepted at remote address /127.0.0.1:35400
Timeout settings for client response taking time: 10000
Timeout settings for server not accepting any client: 10000
<<< hey
>>> ho
Client did not send a message for 10 seconds.
Timeout!
```

```
serhat@ThinkPad: ~/Dev/networking-projects/Project1-1-serhat-coskun-64152$ java Client 8888 localhost
This is the client
Binding to port localhost: 8888
Connected to server at remote address localhost/127.0.0.1:8888
Timeout settings for client time: 0
>>> hey
<<< ho
>>> what
<<< timeout
Can't send the message!
Server timedout because of idleness!
```

3. Networking from different computers

In the following screenshots, chat application running in 1 Mac and 1 Linux machine can be seen. Server is run in the Mac and client on the Linux. All functionality works in this format too. To run the application in different PCs client side should enter the IP address of the server.

```
serhat@ThinkPad:~/Dev/networking-projects/Project1-1-serhat-coskun-64152$ java Client 8888 172.20.62.197
This is the client
Binding to port localhost: 8888
Connected to server at remote address /172.20.62.197:8888
>>> Hallo, meine Freund Kutlu. Wie gets dir?
<<< Hallo bruder! Sech schön, ich liebe dich!
>>> Bruder auf wiedersehen!
<<< tchüss
>>> exit
Exiting the chat application!
serhat@ThinkPad:~/Dev/networking-projects/Project1-1-serhat-coskun-64152$
```

Serhat COŞKUN – 64152
Project1-1

```
Kutluhans-MacBook-Pro:Desktop kutluhanpalalioglu$ javac Server.java
Kutluhans-MacBook-Pro:Desktop kutluhanpalalioglu$ java Server.java 8888
This is the server
Listening to port: 8888
Client accepted at remote adress /172.20.44.61:33938
<<< Hallo, meine Freund Kutlu. Wie gets dir?
>>> Hallo bruder! Sech schön, ich liebe dich!
<<< Bruder auf wiedersehen!
>>> tchüss
<<< exit
Exitting the chat application!
Kutluhans-MacBook-Pro:Desktop kutluhanpalalioglu$
```