

Network Programming Project

This is a simple network-based chat application in Java, which uses the Java Socket API. It consists of two applications: ClientSide and ServerSide.

Basic Functionality:

The basic functionality of this program is as follows:

On the ServerSide application, to begin the user runs the application by entering the port number to the command line i.e.

```
java -jar ServerSide.jar 11.
```

The Server starts up and waits for the socket connection to the port the user entered.

On the ClientSide, the user enters the portnumber followed by the hostname. i.e. `java -jar Client.jar 11 localhost`.

The Client starts up and attempts to create a socket connection to the server.

When the connection is successfully created, the server sends the first message to the client and the client listens to the socket and prints the message.

The Client application can send a message over to the ServerSide and this process repeats. In some cases the client or server cannot send a message across until they have received a message from the other side. This is because the call to listen to the other side as well as the call that reads the user input are both blocking.

Users can end the chat session by entering `"/q"` into the console.

When the user types `"/q"` the console prints `"You have exited the chat"` and the message printed to the other application is `"Lost connection to the server."`

If the Client application enters the wrong portnumber at the command line, the console will print `"Failed to connect to chat"`.

Running Instructions:

In order to run the application:

1. In the ServerSide application enter `java -jar ServerSide.jar` followed by the portnumber you wish to connect to i.e. `java -jar ServerSide.jar 11`.
2. When it connects it will print `"Welcome to the chat"`. The user can now enter a message and send it to the Client application.
3. In the Client application enter `java -jar Client.jar` followed by the portnumber and the hostname you wish to connect to i.e. `java -jar Client.jar 11 localhost`.
4. When it connects it will print `"Welcome to the chat"` followed by a message from the ServerSide depending on whether the user has sent a message from the ServerSide application

yet. If the Client has already received a message from the ServerSide application, the user can now enter a message and send it to the ServerSide application from the Client.

5. To quit the chat a user can enter `/q` on either the Client or ServerSide. When the user enters this the chat will print "You have exited the chat." If the user in the SeverSide enters `/q`, the Client will receive the message "Lost connection to server." If the user in the Client presses `/q`, the ServerSide will print "Client disconnected".

Consulted Sources

Initially the code from the Daytime Client and DaytimeServer apps were consulted in order to form the connection. The code was then reformed and added to in order to achieve the required functionality for this project.