

# Chat 9000 Client/Server application Readme

## Overview

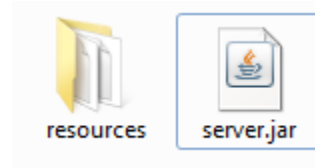
Chat 9000 is a simple Java based chat application that allows communication between multiple parties by means of secure socket connection. It supports both private and public messages to selected parties. Please note that this application is not intended for sending long passages (more than 80 characters at a time) of text nor file transferring.

## Launching

Launching procedure for client and server applications is identical in both Windows and Linux operating systems. Both client and server are supplied in .tar archives. To extract the archives please run **tar -xvzf \*.tar** where \* is the tar filename. Please make sure to start the server before attempting to connect any clients.

### Launching Server

- Before launching the server please make sure that the server jar is located in the same folder as the **resources** folder with all its contents supplied with the jar.



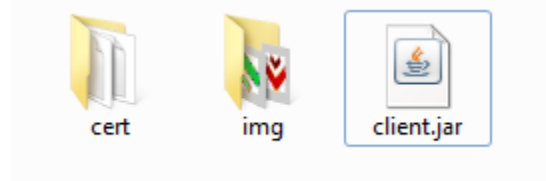
- The application has been developed and tested using **JRE1.6**, thus please make sure that your JRE version is equivalent or greater.
- To launch the application use the command:

```
java -jar server.jar
```

Figure 1 Launching server

### Launching Client

- Before launching the client please make sure that the client jar is located in the same folder as the **cert** and **img** folders with all their contents as supplied in the jar.



- The application has been developed and tested using **JRE1.6**, thus please make sure that your JRE version is equivalent or greater.
- To launch the application use the command:

```
java -jar client.jar
```

Figure 2 Launching client

## Using client

When the client is first launched the following login dialog appears:

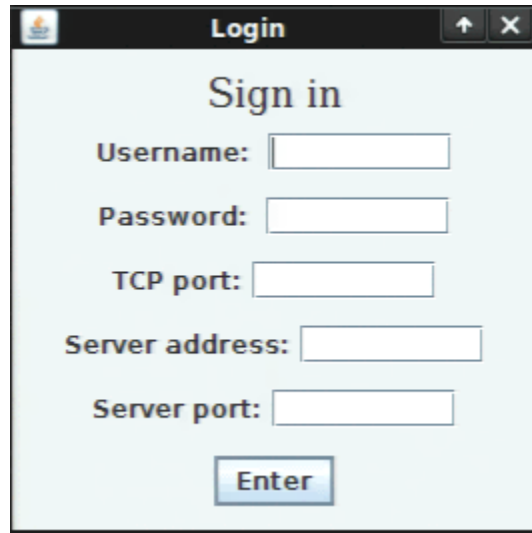
A screenshot of a 'Login' dialog box. The title bar says 'Login' with a small icon on the left and standard window controls on the right. The main area has a light blue background and is titled 'Sign in'. It contains five input fields: 'Username:', 'Password:', 'TCP port:', 'Server address:', and 'Server port:'. Each field is followed by a text input box. At the bottom center is a button labeled 'Enter'.

Figure 3 Authentication dialog

At this point you have to provide the following parameters:

- **Username** - username of the user that you want to login as. (Necessary)
- **Password** - his or her corresponding password. (Necessary)
- **TCP port** - local port to be used by the client. (Optional)
- **Server address** - properly formatted IP address (or alias) of the server that we're trying to connect to. Usually "loopback". (Necessary)
- **Server port** - port on which the target server is running (50000 by default). (Necessary)

This is a list of usernames and passwords for the application:

1. dima - congratulationsClassmate:)
2. tommy - thisoneiseasy
3. lena - cat1985ip
4. jack - thePirateOC
5. tim - th1sguy1s1mp0ss1bL3

Once you have clicked enter, the authentication sequence occurs, and if it is successful - you are logged in:

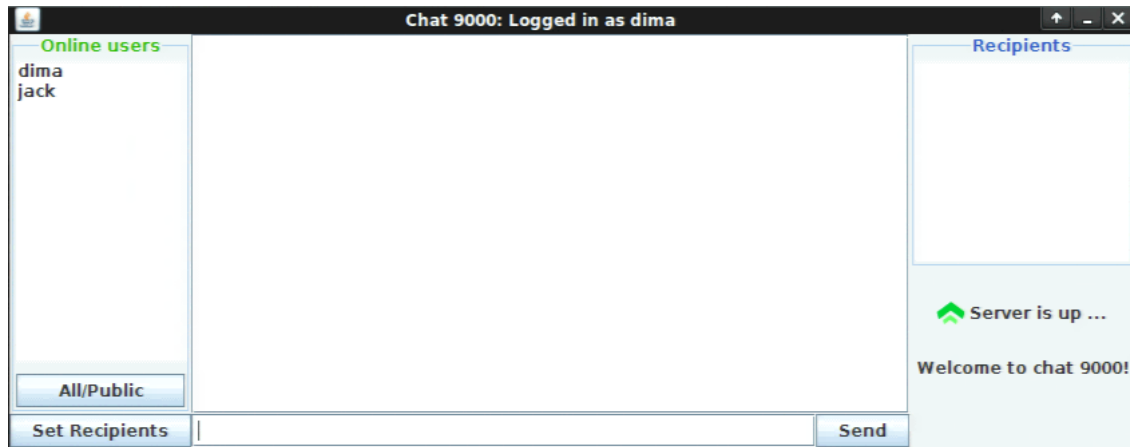
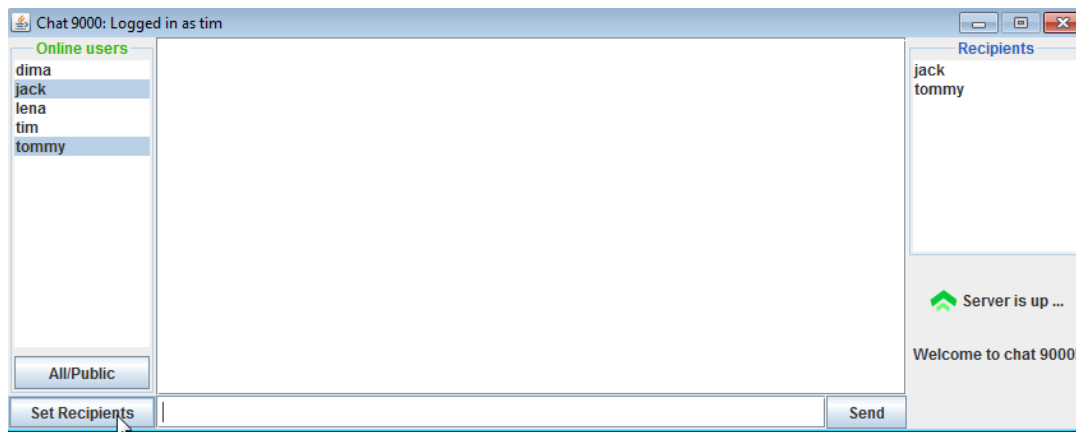


Figure 4 Main screen

In the main screen of the chat the following can be done:

- Select recipients in the **Online Users** list by holding **CTRL** while clicking on the users. Then click **Set Recipients** to set them as current recipients of the messages. Or click **All/Public** to start sending messages to everyone.



- Type messages in the bottom textbox and click **Send** or **press Enter** to send messages.

You are now able to send and receive messages to and from other users.

## Using server

While there are no modifications that can be done to the server in run time - there are a couple of files that can be configured before launching the application:

- **server.properties** - is used to configure the port (PORT\_NUMBER) on which the application server runs. Standard Java properties notation apply: name and value of the attribute are separated by '='.

```
#Port number of the server  
PORT_NUMBER = 50000
```

- **accounts.data** - is used to keep track of the user accounts. It does not contain actual names and passwords, but their hashed representations instead. Format of the file is as follows **username <tab> id <tab> password**. To add users the administrator has to calculate the SHA-512 hash of the new user and password and insert the new tuple in the file. To remove a user the hash of the username must be found and the corresponding tuple removed. Modification is done similarly.

For more information consult the presentation.