

Group Members: Shashank Seeram,Amar Bakir

Internet Technology

Professor Krzyzanowski

Spring 2016

Compiling Instructions

On command line, compile Server.java file using the command `javac Server.java`. Once its compiled, you may run the Server with arguments with the command `java Server -p [port num]`. This is to specify a customized port number. After you have the Server running, you should compile both, Post.java and Get.java with the commands `javac Post.java` and `javac Get.java` respectively. For Post.java, you can run it with arguments with the following command `java Post [-h hostname] [-p port] groupname` and `java Get [-h hostname] [-p port] groupname` for Get.java. You should also compile message.java and group.java. After running Post.java, the program will read messages from client and send it to the Server which then stores it in a hash map. Get.java when given a valid group name will send a request to the Server to get all the messages stored in the hash map which will then be returned to the client.

We decided to use a hash map to store all the messages in each group where group is a key and the value is a linked list of all messages in that specific group. This we figured is the most efficient and easiest way to store all the messages. Every key is unique, meaning there cannot be any duplicates in terms of group names. There are no bugs or peculiarities and everything should match exactly with the way you wanted the formatting to be. A little issue seems to be with our spacing, you have specified in the assignment instructions that there should be a blank line after printing out the message in Get.java only if there is another message after it but our program seems to always print out a blank line. This may be something related to the console and we are not very sure. (Although we do have a check for it in our code!)

Test Cases:

Test#1:

```
javac Server.java
java Server -p 11111
```

```
javac Post.java
javac Get.java
java Post -p 11111 testgroup
```

/*The server will return 'ok' response to which the client will create a new group called 'testgroup' and record all the messages that the client enters*/

This is a test message

*/*Sends this message to Server*/*

java Get testgroup

1 messages

From username /192.168.60.142:58713 Fri Feb 05 16:13:14 EST 2016

This is a test message

Test #2:

java Post test group

error: invalid groupname

Test #3:

java Post -p 65536

error: invalid command

Test #4:

After having the server running, you should open two terminal windows to run 2 or more clients to check if multi-threading works. This should work perfectly fine.

Test #5:

Entering an invalid groupname for get will return an error.

Test #6:

Posting to a group that doesn't exist returns an error.

Test #7:

Entering wrong port numbers, illegal group names, and extra arguments will all return errors