Sean Jergensen

Patrick Dickey

PT2

Testcase 1001

System: Simple Chat Phase: 3

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console C, enter: java ClientConsole user2

4. At console B, enter: #block user2

5. At console C, enter: Hello

Expected result:

1. Console B should report it is blocking messages from user2 and after user2 sends a message nothing should be present Console B, but it should be present on Console 3.

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1002

System: Simple Chat Phase: 3

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console C, enter: java ClientConsole user2

4. At console B, enter: #block user2

5. At console C, enter: #private user1 hello

Expected result:

1. Console B should report it is blocking messages from user2 and after user2 sends a message nothing should be present Console B even if the message is private. Console C should say: Cannot send message because user1 is blocking you.

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1003

System: Simple Chat Phase: 3

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console C, enter: java ClientConsole user2

4. At console B, enter: #block user2

5. At console C, enter: #forward user1

Expected result:

1. Console B should report it is blocking messages from user2.

2. Console C should report: “Cannot forward to user1 because user is blocking messages from you.”

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1004

System: Simple Chat Phase: 3

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console C, enter: java ClientConsole user2

4. At console D, enter: java ClientConsole user3

5. At console B, enter: #block user3

6. At console C, enter: #forward user1

7. At console D, enter: #private user2 Hello

Expected result:

1. Console B should report it is block user3

2. Console C should report it is forwarding messages to user1

3. Console B should report it is receiving forwarded messages from user 2

4. Console B should not receive the private message sent by user3

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1005

System: Simple Chat Phase: 3

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console B, enter: #block user2

4. At console C, enter: java ClientConsole user2

5. At console C, enter: #whoblockme

Expected result:

1. Console B should report it is blocking messages from user2.

2. Console C should report user1 1 is blocking them

Cleanup:

Hit CTRL+C to kill the server.

Testcase 1006

System: Simple Chat Phase: 3

Tests basic functionality of private messages

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console C, enter: java ClientConsole user2

4. At console B, enter: #private user2 Test Message

Expected result:

1. Console C should display: user1> (Private) Test Message

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1007

Test sending private message to yourself

System: Simple Chat Phase: 3

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console B, enter: #private user1 Test Message

Expected result:

1. Console B should display: user1> (Private) Test Message

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1008

Tests sending private message to nonexistent user

System: Simple Chat Phase: 3

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console B, enter: #private MadeUpUser Test Message

Expected result:

1. Console B should display: ERROR- User does not exist

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1009

Tests normal meeting monitoring

System: Simple Chat Phase: 3

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console C, enter: java ClientConsole user2

4. At console D, enter: java ClientConsole user3

5. At console B, enter: #meeting user2

6. At console D, enter: Test Message

Expected result:

Console B should display:

In meeting: user2 will now receive your messages. When you return type #endmeeting to cancel forwarding.

Console C should display:

user1> user1 is in a meeting and has selected you to monitor their chat. You will now receive all of user1's messages

user3> Test Message

user1> user3> Test Message

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1010

System: Simple Chat Phase: 3

Test setting meeting monitor to nonexistent user

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console B, enter: #meeting MadeUpUser

Expected result:

1. Console B should display: ERROR - User to monitor chat must exist.

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1011

System: Simple Chat Phase: 3

Test setting yourself as the meeting monitor

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console B, enter: #meeting user1

Expected result:

1. Console B should display: ERROR - You cannot monitor your own chat.

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1012

System: Simple Chat Phase: 3

Basic channel usage tests

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console C, enter: java ClientConsole user2

4. At console B, enter: Initial Channel Test Message

5. At console B, enter: #setchannel Test

6. At console B, enter: Test Message From Different Channel

7. At console C, enter: #setchannel Test

6. At console B, enter: Test Message From Same Channel

Expected result:

1. Console B should display:

user1> Initial Channel Test Message

Channel has been set to: Test

user1> Test Message From Different Channel

user1> Test Message From Same Channel

2. Console C should Display:

user1> Initial Channel Test Message

Channel has been set to: Test

user1> Test Message From Same Channel

Cleanup:

Hit CTRL+C to kill the server and clients.

Testcase 1013

System: Simple Chat Phase: 3

Ensures 2 clients cannot login using same username to impersonate one another.

Instructions:

1. At console A, enter: java EchoServer

2. At console B, enter: java ClientConsole user1

3. At console C, enter: java ClientConsole user1

Expected result:

2. Console C should Display:

ERROR- A user with the id: user1 is already online. Awaiting Command

Cleanup:

Hit CTRL+C to kill the server and clients.