

CS314 Simplechat P2 Testcases

Caleb Tebbe & Zach Kaplan

April 1, 2013

1 Testcase D1001

- System: SimpleChat
- Phase: 3
- Description: Test private messages between clients.

Rationale

Test to see if a client has the ability to send a private message to another connected client.

Instructions

1. Start server
2. Start a client logged in as test
3. Start a client logged in as test2
4. Start a client logged in as test3
5. Use command “#private test2 test message” from client test

Expected Results

1. Client test2 will receive private message “test message”
2. Client test3 will not see any messages

Clean up

1. Hit CTRL+C to kill any remaining clients/servers ...

2 Testcase D1002

- System: SimpleChat
- Phase: 3
- Description: Test creation and use of individual channels.

Rationale

Test to see if clients can create and join channels to send messages through the channel.

Instructions

1. Start server
2. Start a client logged in as test
3. Start a client logged in as test2
4. Start a client logged in as test3
5. Run client command from test: “#channel newchan”
6. Run client command from test2: “#channel newchan”
7. Run client command from test2: “#channel newchan test message”

Expected Results

1. Client test will create channel *newchan*
2. Client test2 will join existing channel *newchan*
3. Client test will receive channel message “test message”
4. Client test3 will not see any messages

Clean up

1. Hit CTRL+C to kill any remaining clients/servers ...

3 Testcase D1003

- System: SimpleChat
- Phase: 3
- Description: Test client forwarding functionality.

Rationale

Test to see if client can set up forwarding to another client. Forwarding should override any client blocking or channel restrictions by the receiving client.

Instructions

1. Start server
2. Start a client logged in as test
3. Start a client logged in as test2
4. Start a client logged in as test3
5. Run client command from test: “#channel newchan”
6. Run client command from test2: “#channel newchan”
7. Run client command from test3: “#block test2”
8. Run client command from test: “#forward test2”
9. Run client command from test2: “#channel newchan test channel message”
10. Run client command from test2: “#private test test private message”

Expected Results

1. Client test will create channel *newchan*
2. Client test2 will join existing channel *newchan*
3. Client test3 will block *test2*
4. Client test will start forwarding to *test2*
5. Client test will receive channel message “test channel message”
6. Client test will receive private message “test private message”
7. Client test3 will receive forwarded channel message “test channel message”
8. Client test3 will receive forwarded private message “test private message”

Clean up

1. Hit CTRL+C to kill any remaining clients/servers . . .

4 Testcase D1004

- System: SimpleChat
- Phase: 3
- Description: Test client available/unavailable functionality.

Rationale

Test to see if a client can become *#unavailable* to ignore any messages and later use *#available* to get messages again.

Instructions

1. Start server
2. Start a client logged in as test
3. Start a client logged in as test2
4. Start a client logged in as test3
5. Run client command from test: “#channel newchan”
6. Run client command from test2: “#channel newchan”
7. Run client command from test2: “#channel newchan test channel message”
8. Run client command from test2: “#private newchan test private message”
9. Run client command from test: “#unavailable”
10. Run client command from test2: “#channel newchan test channel message 2”
11. Run client command from test2: “#private test test private message 2”
12. Run client command from test: “#available”
13. Run client command from test2: “#channel newchan test channel message 3”
14. Run client command from test2: “#private test test private message 3”

Expected Results

1. Client test will create channel *newchan*
2. Client test2 will join existing channel *newchan*
3. Client test3 will block *test2*

4. Client test will receive channel message “test channel message”
5. Client test will receive private message “test private message”
6. Client test will become *unavailable*
7. Client test will **not** receive channel message “test channel message 2”
8. Client test will **not** receive private message “test private message 2”
9. Client test2 will receive message “Client test is unavailable”
10. Client test will become *available*
11. Client test will receive channel message “test channel message 3”
12. Client test will receive private message “test private message 3”
13. Client test3 will not receive any messages

Clean up

1. Hit CTRL+C to kill any remaining clients/servers ...

5 Testcase D1005

- System: SimpleChat
- Phase: 3
- Description: Test client blocking feature.

Rationale

Test to see if clients can block messages from other clients/server.

Instructions

1. Start server
2. Start a client logged in as test
3. Start a client logged in as test2
4. Start a client logged in as test3
5. Run client command from test: “#channel newchan”
6. Run client command from test2: “#channel newchan”
7. Run client command from test2: “#channel newchan test message”
8. Run client command from test2: “#private newchan test private message”
9. Send server message: “test server message”
10. Run client command from test: “#block test2”
11. Run client command from test: “#block server”
12. Run client command from test2: “#channel newchan test channel message 2”
13. Run client command from test2: “#private test test private message 2”
14. Send server message: “test server message 2”

Expected Results

1. Client test will create channel *newchan*
2. Client test2 will join existing channel *newchan*
3. Client test will receive channel message “test message”
4. Client test will receive private message “test message”
5. Client test will receive server message “test server message”

6. Client test will block user *test2*
7. Client test will **not** receive channel message “test channel message 2”
8. Client test will **not** receive private message “test private message 2”
9. Client test will **not** receive server message “test server message 2”
10. Client test3 will only see server messages

Clean up

1. Hit CTRL+C to kill any remaining clients/servers ...

6 Testcase D1006

- System: SimpleChat
- Phase: 3
- Description: Test server blocking feature.

Rationale

Test to see if server can block messages from clients.

Instructions

1. Start server
2. Start a client logged in as test
3. Send client message: “test message”
4. Run server command: “#block test”
5. Send client message: “test message 2”

Expected Results

1. Server will receive client message “test message”
2. Server will block client *test*
3. Server will **not** receive client message “test message 2”

Clean up

1. Hit CTRL+C to kill any remaining clients/servers ...

7 Testcase D1007

- System: SimpleChat
- Phase: 3
- Description: Test impostor detection.

Rationale

Test to see if server can detect multiple users signing in with the same login-id

Instructions

1. Start server
2. Start a client logged in as client1
3. Start a client logged in as client1

Expected Results

1. The first client will log in with ID *client1*
2. Server will respond to second client with an error that there is an existing client with loginID *client1*
3. Server will notify the client that is attempting to connect that the ID *client1* is already in use

Clean up

1. Hit CTRL+C to kill any remaining clients/servers ...