Networking

Commands that will be sent to the server will begin with a “/” to execute a command. The user must have a user name before using commands with the exception of the username and login commands.

Commands will be written in this format

**Command Name**

Function: Commands function

Client Inputs: /command <command parameter>

Restrictions: Input Restrictions

Valid server response: Server response

Invalid server response: Server response

## Command List:

**Username**

Function: Sets the users name that will be seen by other users as long as the name is not taken.

Client Inputs: /user <username>

Restrictions: No symbols such as @, !, / and Spaces

Valid server response: “USER\_OK”

Invalid server response: “ERR\_USER\_TAKEN”, “ERR\_INVALID\_NAME”

**Set Password**

Function: Allows users to set a password after they have set a unique username.

Client Inputs: /setpass <password>

Restrictions: No spaces

Valid server response: “USER\_PASS\_SET”

Invalid server response: “ERR\_INVALID\_PASS”

**Login**

Function: Allows users with passwords to log in to their account.

Client Inputs: /login <username> <password>

Restrictions: Same restrictions apply as they do with the Username and Set Password commands.

Valid server response: “USER\_LOGIN\_OK”

Invalid server response: “ERR\_LOGIN\_FAIL”, “ERR\_INVALID\_SYNTAX”

**Log Out**

Function: Logs a user out.

Client Inputs: /logout

Restrictions: N/A

Valid server response: “USER\_LOGOUT\_OK”

Invalid server response: N/A

**Open Rooms**

Function: See public rooms by name that are not full

Client Inputs: /openrooms

Restrictions: No private rooms or full public rooms.

Valid server response: “USER\_OPEN\_ROOMS”, Returns with list of open rooms separated by commas.

Invalid server response: N/A

**Search For Rooms**

Function: Search for rooms by name or tag

Client Inputs: /search <room name> or /search <tag>

Restrictions: Same restrictions apply as they do with the Create Room command.

Valid server response: “USER\_SEARCH”, Returns with list of open rooms separated by commas and sorted by room name or tag.

Invalid server response: “ERR\_ROOM\_NOT\_FOUND”, ”ERR\_TAG\_NOT\_FOUND”, “ERR\_INVALID\_SYNTAX”

**Create Room**

Function: create a room that is private or public and become room leader

Client Inputs: /create <privacy> <room name>

Restrictions: Room name may not contain spaces or symbols

Valid server response: “ROOM\_PUBLIC”, “ROOM\_PRIVATE”

Invalid server response: “ERR\_ROOM\_TAKEN”, “ERR\_INVALID\_SYNTAX”

**Close Room**

Function: Closes a room

Client Inputs: /close <room name>

Restrictions: Must be a room leader to close a room

Valid server response: “ROOM\_CLOSE”

Invalid server response: “ERR\_ROOM\_NOT\_FOUND”

**Join Room**

Function: Joins a public room

Client Inputs: /join <room name>

Restrictions: Room name must exist.

Valid server response: “USER\_JOIN\_ROOM”,

Invalid server response: “ERR\_ROOM\_FULL”, “ERR\_ROOM\_NOT\_FOUND”

## Room specific commands:

**Add Tag**

Function: Add a tag to a room so it is easier for others to find

Client Inputs: /tag <room name> <tag>

Restrictions: Room name must exist. Tags may no have spaces or symbols. Must be room leader to add tags.

Valid server response: “ROOM\_TAGGED”

Invalid server response: “ERR\_ROOM\_NOT\_FOUND”, “ERR\_INVALID\_SYNTAX”

**Remove Tag**

Function: Remove a tag from a room

Client Inputs: /untag <room name> <tag>

Restrictions: Room name must exist. Tag must exist and have no spaces. Must be room leader to remove tags.

Valid server response: “ROOM\_UNTAGGED”

Invalid server response: “ERR\_ROOM\_NOT\_FOUND”, “ERR\_INVALID\_SYNTAX”, “ERR\_TAG\_NOT\_FOUND”

**Leave Room**

Function: Leave a room and return to Lobby

Client Inputs: /leave <room name>

Restrictions: Room name must exist.

Valid server response: “USER\_LEAVE\_ROOM”

Invalid server response: “ERR\_ROOM\_NOT\_FOUND”

**Promote User**

Function: Promote a user in a room to moderator

Client Inputs: /promote <username>

Restrictions: Same restrictions apply as they do with the Username command.

Valid server response: “USER\_PROMOTE\_MOD”

Invalid server response: “ERR\_ROOM\_NOT\_FOUND”, “ERR\_INVALID\_NAME”

**Demote User**

Function: Demote a user from room moderator

Client Inputs: /demote <username>

Restrictions: Must be a room leader to demote a user and that user must be a moderator to be demoted.

Valid server response: “USER\_DEMOTE”

Invalid server response: “ERR\_INVALID\_NAME”, “ERR\_INVALID\_USER”

**Kick User**

Function: Remove a user from the room

Client Inputs: /kick <username>

Restrictions: Must be a room leader or a moderator to kick a user

Valid server response: “USER\_KICK”

Invalid server response: “ERR\_INVALID\_NAME”

**Ban User**

Function: Permanently remove a user from the room

Client Inputs: /ban <username>

Restrictions: Must be a room leader to ban a user

Valid server response: “USER\_BAN”

Invalid server response: “ERR\_INVALID\_NAME”

**Room Settings**

Function: Set rules for the room

Client Inputs: /set <room setting> <setting parameter (if applicable)> <state>

Restrictions: Must be a room leader to activate or disable room settings. State must be true or false. Room settings include; moderator only chat, word limit, user limit.

Valid server response: “ROOM\_SET\_RULE”

Invalid server response: “ERR\_INVALID\_SETTING”, “ERR\_INVALID\_SYNTAX”

**Invite User**

Function: Invites a user to a private room

Client Inputs: /invite <username>

Restrictions: Username must exist.

Valid server response: “USER\_INVITE\_SENT”

Invalid server response: “ERR\_INVALID\_USER”

**Accept Invite**

Function: Accepts an invite from user

Client Inputs: /accept <username>

Restrictions: Must have an existing invite. And username must exist

Valid server response: “USER\_INVITE\_ACCEPT”

Invalid server response: “ERR\_INVALID\_USER”

**Private Message User**

Function: Allows for one user to send a private message to another

Client Inputs: /pm <username> <text>

Restrictions: Username must exist.

Valid server response: “PRIVATE\_MESSAGE\_SENT”

Invalid server response: “ERR\_INVALID\_NAME”, “ERR\_INVALID\_SYNTAX”

**Users in room**

Function: Shows user the users in a room

Client Inputs: /whosin <room>

Restrictions: Room must exist.

Valid server response: “USER\_IN\_ROOM”, Returns with list of users in a room separated by commas.

Invalid server response: “ERR\_INVALID\_NAME”, “ERR\_INVALID\_SYNTAX”

## Text manipulation commands:

To use a text manipulation command the user must start with the command and then a “/” and end it with “/”

These commands will be written in this format

**Command Name**

Function: Commands function

Client Inputs: /command/<text>/

Restrictions: Input Restrictions

Invalid server response: Server response

**Bold Text**

Function: Bold text between the slashes.

Client Inputs: /bold/<text>/

Restrictions: Cannot use a forward slash in the text.

Invalid server response: “ERR\_INVALID\_TEXT”, ”ERR\_INVALID\_SYNTAX”

**Italics Text**

Function: The text between the slashes become italicised

Client Inputs: /italic/<text>/

Restrictions: Cannot use a forward slash in the text.

Invalid server response: “ERR\_INVALID\_TEXT”, ”ERR\_INVALID\_SYNTAX”

**Underline Text**

Function: The text between the slashes become underlined

Client Inputs: /underline/<text>/

Restrictions: Cannot use a forward slash in the text.

Invalid server response: “ERR\_INVALID\_TEXT”, ”ERR\_INVALID\_SYNTAX”

**Colour Text**

Function: The text between the slashes become a colour specified by the command

Client Inputs: /c <colour> <text> or /c <red> <green> <blue> <text>

Restrictions: colour must be from standard colours and have no symbols numbers or spaces. Red, green and blue must have a value between 0 to 255 and must not contain any letters, symbols or spaces.

Invalid server response: “ERR\_INVALID\_TEXT”, ”ERR\_INVALID\_SYNTAX”

## Miscellaneous Sever Responses

When the user logs in they are prompted to use either the username or login command.

Example: “Hello, please use /user to assign a name for yourself or /login if you already have an account”