Chat Protocol Design

Contents

[General notes: 2](#_Toc35198037)

[Packet Design 2](#_Toc35198038)

[Responses 2](#_Toc35198039)

[Access Design: 3](#_Toc35198040)

[Client sent commands: 4](#_Toc35198041)

[Server sent commands: 6](#_Toc35198042)

[Server Response Values: 7](#_Toc35198043)

## General notes:

|  |  |  |
| --- | --- | --- |
| Network Protocol | TCP |  |
| Server IP | I cannot promise this will be static |  |
| Port | 1990 |  |
| Network Data Packet design | XML |  |

### Packet Design

XML has been chosen as the design of the packets. To explain how this would look, a packet for a SEND\_MESSAGE command would look like the following:

<SEND\_MESSAGE><message>FooBar</message></SEND\_MESSAGE>

This approach was chosen over other alternatives, such as comma delimited, as those restrict the most common punctuation used by users, the comma. This would be a problem with normal text as well as ASCII art sent over the protocol.

Using XML, however, provides a universal standard for how the data should be packaged. As for ease of use, XML is so widely adopted there are many third-party tools for easily assembling and dissembling XML data.

Another pro of XML is data checking. Since it uses both an opening brace <SEND\_MESSAGE> and a closing brace </SEND\_MESSAGE>, we can make sure we have received the full message because we should have all the braces. If any are missing, we have not received the full message and that can be handled. While this can be done with the alternatives, that would be a ‘home-brew’ method whereas this is in the XML standard as default.

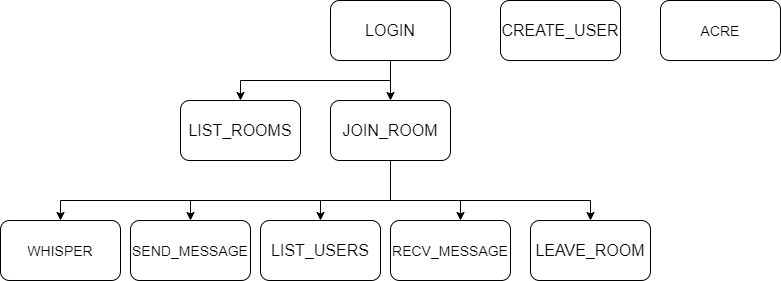
### Responses

This is a brief of what responses from the server are and how they are sent to the client.

In the ‘Client Sent Commands’ section, each table has a row for ‘Response’. This is the response sent by the server to acknowledge this command has been processed, and of any issues encountered whilst processing the command. These responses are sent to the client by the ‘ACRE’ command explained in ‘Server Sent Commands’.

## Access Design:

This image serves as a reference to determine the nature of the ERR\_ACCESS response sent by the server. Each level down represents commands unavailable to the user until they have satisfied a new state.

For example, a user cannot use the LIST\_ROOMS or JOIN\_ROOM commands until they have logged in by using the LOGIN command.

## Client sent commands:

These are the commands sent BY the client, TO the server.

|  |  |
| --- | --- |
| Function name | Join Room |
| Syntax | JOIN\_ROOM <roomName> |
| Definition | A request for a user to join a specific room by name.   |  |  |  | | --- | --- | --- | | Name | Type | Description | | roomName | String | the name of the room to join, may not contain spaces | |
| Response | ERR\_ROOMINVALID  ERR\_ROOMFULL  ERR\_NEEDMOREPARAMS  ERR\_TOOMANYPARAMS  ERR\_ACCESS  PSS\_OK |

|  |  |
| --- | --- |
| Function name | Leave Room |
| Syntax | LEAVE\_ROOM |
| Definition | A request for a user to leave their current room.  Due to users being restricted to one room at a time, there is no need for any parameters. |
| Response | ERR\_TOOMANYPARAMS  ERR\_NEEDMOREPARAMS  ERR\_ACCESS  PSS\_OK |

|  |  |
| --- | --- |
| Function name | Login |
| Syntax | LOGIN <username> <password> |
| Definition | A request for a user to login to the server.   |  |  |  | | --- | --- | --- | | Name | Type | Description | | username | String | The username supplied by the user | | Password | String | The password supplied by the user | |
| Response | ERR\_TOOMANYPARAMS  ERR\_NEEDMOREPARAMS  ERR\_INVALIDLOGIN  PSS\_OK |

|  |  |
| --- | --- |
| Function name | Create User |
| Syntax | CREATE\_USER <username> <password> <passwordcheck> |
| Definition | A request for a user to create a new login for the server.   |  |  |  | | --- | --- | --- | | Name | Type | Description | | username | String | The username supplied by the user | | Password | String | The password supplied by the user | | Passwordcheck | String | A validation check for the user’s password. | |
| Response | ERR\_TOOMANYPARAMS  ERR\_NEEDMOREPARAMS  ERR\_INVALIDLOGIN  PSS\_OK |

|  |  |
| --- | --- |
| Function name | Send Message |
| Syntax | SEND\_MESSAGE <message> |
| Definition | A request for a user to send a message to their current room.   |  |  |  | | --- | --- | --- | | Name | Type | Description | | Message | String | The message written by the user | |
| Response | ERR\_TOOMANYPARAMS  ERR\_NEEDMOREPARAMS  ERR\_ACCESS  PSS\_OK |

|  |  |
| --- | --- |
| Function name | Whisper |
| Syntax | WHISPER <user> <message> |
| Definition | A request for a user to send a message to one user in the same room. Nobody else will receive the message.   |  |  |  | | --- | --- | --- | | Name | Type | Description | | User | String | The user to send the message to | | Message | String | The message written by the user | |
| Response | ERR\_TOOMANYPARAMS  ERR\_NEEDMOREPARAMS  ERR\_ACCESS  PSS\_OK |

|  |  |
| --- | --- |
| Function name | List Rooms |
| Syntax | LIST\_ROOMS |
| Definition | A request for a list of all the active rooms. |
| Response | ERR\_TOOMANYPARAMS  ERR\_ACCESS  PSS\_OK |

|  |  |
| --- | --- |
| Function name | List Users |
| Syntax | LIST\_USERS |
| Definition | A request for a list of all the users in the user’s current room. |
| Response | ERR\_TOOMANYPARAMS  ERR\_ACCESS  PSS\_OK |

## Server sent commands:

These are the commands sent BY the server, TO the client.

|  |  |
| --- | --- |
| Function name | Receive Message |
| Syntax | RECV\_MESSAGE <message> <user> |
| Definition | Sent from the server to all users in a room, telling them of a new message that a user has sent.   |  |  |  | | --- | --- | --- | | Name | Type | Description | | Message | String | The message written by the user | | user | String | The username of the user that sent the message |     Note: this command also encompasses events like users leaving and joining, which will be sent as a message from that user, in order to simplify the protocol.  e.g:  [DANBOB] Wow I love sending messages!  [OSHAWATT123] has left the room. |

|  |  |
| --- | --- |
| Function name | Acknowledge request |
| Syntax | ACRE <response> |
| Definition | Sent from the server to a user when they make a request of any kind to the server. The responses vary and can be seen later in the document.   |  |  |  | | --- | --- | --- | | Name | Type | Description | | Response | Int (enumeration) | The response from the server, explained later in the document | |

## Server Response Values:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Usage | Value |
| ERR\_ROOMINVALID | The room supplied does not exist. | In the JOIN\_ROOM command, the room is checked against active, open rooms. If the room is not found, this error is returned. | 1 |
| ERR\_ROOMFULL | The room is full | In the JOIN\_ROOM command, the room is checked to see if it has reached capacity | 2 |
| ERR\_NEEDMOREPARAMS | The request has been sent with too few parameters | When a user sends a request to the server, the number of parameters received is checked and if this falls short of what is needed, this is sent back | 3 |
| ERR\_TOOMANYPARAMS | The request has been sent with too many parameters | When a user sends a request to the server, the number of parameters received is checked and if that is too many for the given request, this is returned | 4 |
| ERR\_INVALIDLOGIN | The login details sent are incorrect | In the LOGIN command, the username and password are checked against a server-side dictionary. If a match is not found, this is returned. | 5 |
| ERR\_ACCESS | The user can not access the command in their current state. | Certain commands are locked behind being in a certain ‘state’. For example, a user must be in a room in order to use the LIST\_USERS command. See Access Design for more information. | 6 |
| ERR\_INVALIDRGSTR | The user has been declined registration of new user. | This can happen if the passwords sent are different (also checked client-side but checking here for packet corruption) or if the username is taken | 7 |
| PSS\_OK | The request sent to the server was valid, and has been handled correctly | When any command is sent to the server, if there are no errors in the command or the execution of it, this command is returned to let the client know they’re good to move on. | 0 |