Description

Client

The Client module collects data from user. All user input is encoded to JSON via MessageEncoder and then sent to the server. The Client interprets user input by it's state and uses the associated encode function. The Client's states are IDLE (not logged in) and IN_CHATROOM (logged in). Receiving messages from the server is done in a seperate thread, MessageReceiver, which forwards all data to the Client's function: receive_message(…). The Client then inteprets all the messages from MessageReceiver with the module MessageParser, and does the associated actions.

Server

The Server continously runs concurrently to other client handlers to handle incoming new connections: ThreadedTCPServer. When a new connection is detected by the ThreadedTCPServer, a new ClientHandler is created in a seperate thread to handle that single connection. All shared data between the ClientHandlers such as connected_clients and history are stored in static variabled. A ClientHandler responds to new messages by parsing the payload with the module MessageParser, and then encoding a new message to respond to the given client or all clients. When sending messages to all clients, a dictionary (connected_clients) is used to address all clients.