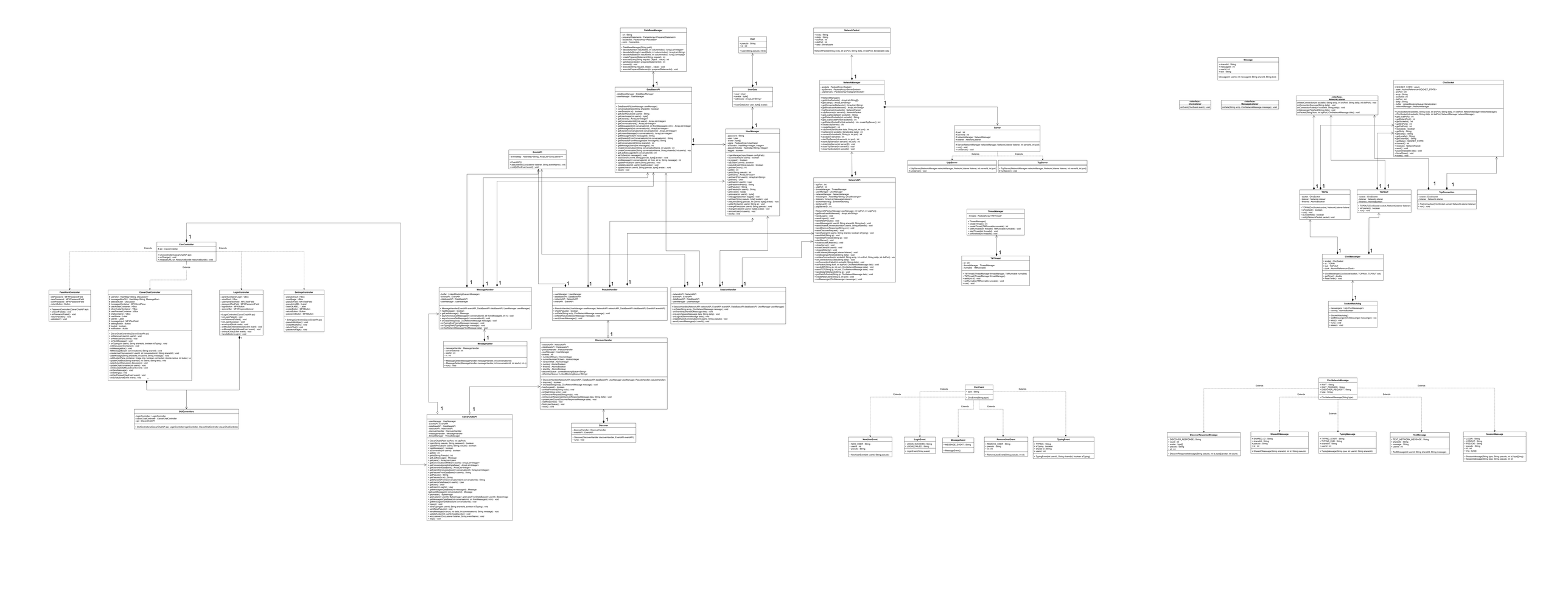
DatabaseManager
- url : String
- preparedStatements : PreparedArrayPreparedStatement>
- resultSet : PreparedArrayResultSet
+ connection : Connection
+ DatabaseManager(String path)
+ deviceExists(int deviceSlot, int columnIndex) : ArrayList<Integer>
+ deviceAttributeExists(int deviceSlot, int columnIndex) : ArrayList<String>
+ getPreparedStatement(int preparedStatementIndex) : ArrayList<PreparedStatement>
+ createPreparedStatement(String request) : int
+ executePreparedStatement(int preparedStatementIndex) : void
+ getIsConnected() : boolean
+ getConnection() : Connection
+ execute(String request, Object ... value) : void
+ execuletPreparedStatement(int preparedStatementIndex) : void
DatabaseAPI
- databaseManager : DatabaseManager
- userManager : UserManager
+ DatabaseAPI(UserManager userManager)
+ connectedUserDeleting event(int, boolean)
+ userExists(int id) : boolean
+ getUserPassword(int id) : String
+ getUserHashed(int userId) : byte[]
+ getUserSalt(int userId) : byte[]
+ getConversationWithUser(int userId) : ArrayList<Integer>
+ getConversationExists(int[] ...) : int
+ getMessagesInConversation(int, int[]ofMessagesId, int n) : ArrayList<Integer>
+ getMessagesInConversation(int, int[]ofMessagesId) : ArrayList<Integer>
+ getMessagesInConversation(int, int[]ofMessagesId, int n) : ArrayList<Integer>
+ getUnreadMessagesIn conversation(int) : ArrayList<Integer>
+ getMessageContent(int messageId) : String
+ getPrivateMsgFromConversationId(int conversationId) : String
+ getMessageAuthorId(int messageId) : int
+ getConversationOf(String, int[]) : int
+ getMessageUserId(int messageId) : int
+ getMessageDate(int messageId, int[]ofMessagesId, int pos0) : int
+ createConversation(String conversationName, String sharedId, int user0) : void
+ addMessageToConversation(int, int, String) : void
+ setOnlineUser(int messageId) : void
+ addMessageIn conversation(int, int[]ofUsers, int n) : void
+ addMessageIn conversation(int, int[]ofUsers) : void
+ updatePrivateUser(int pseudo, String pseudo) : void
+ updateUserPassword(int id, byte[] pass) : void
+ updateUserSalt(int userId, String pseudo, byte[] avatar) : void
+ close() : void
User
+ pseudo : String
+ id : int
+ User(String pseudo, int id)
UserDate
+ user : User
+ avatar : byte[]
+ sharedId : ArrayList<String>
+ UserDate(User user, byte[] avatar)
UserManager
- usersList : String
- user : User
- id : int[]
- users : ParsedArrayUserDate>
- ifFriends : HashMap<Integer, Integer>
- usersMap : HashMap<String, Integer>
- logged : boolean
+ UserManager(ArrayDeque<JsonObject> configPath)
+ getIsUserExists(int pseudo) : boolean
+ isLogged() : boolean
+ isEnabled(int userId) : boolean
+ getUserPrivateKey(int pseudo) : boolean
+ getUserCount() : int
+ getUsersIds() : int[]
+ getId(String pseudo) : int
+ getUserPublicKey(int id) : int
+ getUserPrivateKey(int pseudo) : ArrayList<String>
+ getUser() : User
+ getUsersMap() : Map<String, Integer>
+ getPasswordHash() : String
+ getSalt() : byte[]
+ getPseudo(int userId) : String
+ getPseudo(int pseudo) : String
+ getUsersPrivateKeys() : ArrayList<String>
+ getUsers() : User
+ getUsersList() : String
+ getPasswordHash() : String
+ getSalt() : byte[]
+ getPseudo(int userId) : String
+ getAvatar(int pseudo) : byte[]
+ setUserId(int id) : void
+ setLoggedIn(boolean logged) : void
+ setUserPrivateKey(int id, byte[] pass) : void
+ addUsers(int pseudo, int user0, byte[]avatar) : void
+ addUsers(int pseudo) : void
+ setUsers(int user0) : void
+ reset() : void
NetworkPacket
+ srcIp : String
+ srcPort : int
+ dstPort : int
+ data : byte[]
+ data : Serializable
NetworkPacket(String srcIp, int srcPort, String dstIp, int dstPort, Serializable data)
NetworkManager
- sockets : ParsedArray<Socket>
- socketsMap : HashMap<Integer, Socket>
- clientServer : ParsedArray<DatagramSocket>
+ NetworkManager()
+ getIsSocketSockets() : ArrayList<String>[]
+ getIsClient() : ArrayList<String>
+ getIsSocketSocketsIsClient() : ArrayList<String>
+ getIsSocketsIsClientIsServer() : ArrayList<String>
+ getIsClientIsServer() : ArrayList<String>
+ startNewserver(int servers) : NetworkPacket
+ getSocketSocketsIsClient() : Socket[]
+ getSocketSocketsIsClientIsServer() : String
+ getLoadSocketPortOnt(int socketId) : int
+ getSocketSocketsIsClientIsServer(int socketId) : int
+ closeAll(DatagramSocket[] ...) : void
+ closeAll(Socket[] ...) : void
+ closeAll() : void
+ startNewClient(int socketId, String ip, int port) : int
+ connectOnt(int socketId, String ip, int port) : int
+ connectOnt(int socketId, String ip, int port) : int
+ accept(int servers) : int
+ startSocketPortOnt(int servers, int port) : int
+ startSocketPortOnt(int servers, int port) : int
+ startNewserver(int servers, int port) : int
+ closeTcpServer(int servers) : void
+ closeTcpSocket(int socketId) : void
Server
port : int
servers[] : int
networkManager : NetworkManager
ifClosed : HashMap<Integer, Boolean>
+ Server(NetworkManager networkManager, NetworkListener listener, int servers[], int port)
+ run() : void
+ run(Server) : void
UdpServer
+ UdpServer(NetworkManager networkManager, NetworkListener listener, int servers[], int port)
# run(Server) : void
TcpServer
+ TcpServer(NetworkManager networkManager, NetworkListener listener, int servers[], int port)
# run(Server) : void
NetworkAPI
- tcpPort : int
- udpPort : int
- threadManager : ThreadManager
- networkManager : NetworkManager
- clientsMapper : HashMap<Integer, Client>
- messengers : HashMap<String, ChcMessenger>
- messengersMap : HashMap<Integer, Integer>
- socketsMapping : SocketsMapping
- isClosed : boolean
- isSyncSent() : int
+ NetworkAPI(UserManager userManager, int tcpPort, int udpPort)
+ getBroadcastAddresses() : ArrayList<String>
+ getIsClosed() : int
+ send(byte[] ...) : void
+ sendTcp(byte[] ...) : void
+ sendMessage(int user0, String sharedId, String text) : void
+ sendDiscoverRequest(String ip) : void
+ sendDiscoverResponse(String ip) : void
+ sendSocketRequest() : void
+ sendSyncRequest(int user0, String sharedId, boolean isTyping) : void
+ sendAllMsg(int id) : void
+ sendAllMsg(int[] ...) : void
+ startServer() : void
+ closeServer() : void
+ closeClients() : void
+ closeClient(int socket) : void
+ getClient() : Client
+ addListener(NetworkListener listener) : void
+ getIsClosed() : boolean
+ onNewConnection(int socketId, String ip) : void
+ onNewMessage(int socketId, byte[] data) : void
+ onConnectionFailed(int socketId) : void
+ onDisconnect(int socketId, String ip) : void
+ onLogin(ChcMessenger messenger) : void
+ onLogoutDiscover(String data) : void
+ onNewMessage(int socketId, byte[] data, int offset, int length) : void
+ onNewDiscoverResponse(String data) : void
+ onNewSocketRequest(int socketId, String pseudo) : void
+ sendUserAllMessagesIn user(int, int) : void
+ onNewMessagesChcMessenger(ChcMessenger messenger) : void
ThreadManager
- threads : ParsedArray<TMThread>
+ ThreadManager()
+ create(Thread[] ...)
+ addRunnable(TMRunnable runnable) : void
+ addRunnable(int threadId, TMRunnable runnable) : void
+ stopThread(int threadId) : void
+ isFinished(int threadId) : int
TMThread
- id : int
- threadManager : ThreadManager
- runnable : TMRunnable
+ TMThread(ThreadManager threadManager, TMRunnable runnable)
+ TMThread(ThreadManager threadManager)
+ setId(int id) : void
+ setRunnable(TMRunnable runnable) : void
+ run() : void
Message
+ sharedId : String
+ messageId : int
+ userId : int
+ text : String
Message(int userId, int messageId, String sharedId, String text)
<<interface>>
ChcListener
onExecuteCmd(Cmd event) : void
<<interface>>
MessageListener
onDataString ip, ChcNetworkMessage message) : void
<<interface>>
NetworkListener
onNewConnection(int socketId, String ip) : void
onNewMessage(int socketId, byte[] data, int offset, int length) : void
onConnectionFailed(int socketId) : void
onDisconnect(int socketId, String ip) : void
onPacketReceived(NetworkPacket packet) : void
onPacketSent(String from, int toPort, ChcNetworkMessage data) : void
ChcSocket
+ SOCKET_STATE : enum
+ ENUM_SOCKET_STATE{NotSocket(SOCKET_STATE)}
+ socketId : int
+ ip : String
+ port : int
+ isClosed : int
+ isConnected : int
+ buffer : LinkedBlockingQueue<SerializableInt>
+ networkManager : NetworkManager
+ ChcSocket(int socketId, String ip, int port, int isClosed, int isConnected, NetworkManager networkManager)
+ ChcSocket(int socketId, String ip, int port, int dstPort, int isClosed, int isConnected, NetworkManager networkManager)
+ getLastPacket() : int
+ getIsClosed() : int
+ getSocketId() : int
+ getIsIp() : int
+ getIsPort() : int
+ getIsConnected() : int
+ setIsClosed() : boolean
+ setIsClosed() : void
+ getIsIp() : String
+ getIsPort() : int
+ getSocketIp() : String
+ getSocketPort() : int
+ getSocketId() : int
+ setSocketId() : void
+ setSocketIp() : void
+ setSocketPort() : void
+ setIsIp() : void
+ setIsPort() : void
+ setIsConnected() : void
+ setIsClosed() : void
+ setIsIp() : void
+ setIsPort() : void
+ setIsConnected() : void
+ setIsClosed() : void
+ setSocketIp() : void
+ setSocketPort() : void
+ setSocketId() : void
+ receive() : NetworkPacket
+ close() : void
+ putDatagrams(byte data) : void
+ getDatagrams() : byte[]
+ close() : void
TcpConnection
- socket : ChcSocket
- networkManager : NetworkManager
- listener : NetworkListener
+ TcpConnection(ChcSocket socket, NetworkListener listener)
+ run() : void
TCPIn
- socket : ChcSocket
- networkManager : NetworkManager
- listener : NetworkListener
+ TCPIn(ChcSocket socket, NetworkListener listener)
+ ifFinished() : boolean
+ run() : void
+ isCloseWait() : boolean
+ onNewPacketReceived(packet) : void
TCPOUT
- socket : ChcSocket
- networkManager : NetworkManager
- listener : NetworkListener
+ TCPOUT(ChcSocket socket, NetworkListener listener)
+ ifFinished() : boolean
+ run() : void
ChcMessenger
- socket : ChcSocket
- socketIn : TCPIn
- out : TCPOUT
- socket : AtomicReference<Client>
+ ChcMessenger(ChcSocket socket, TCPIn in, TCPOUT out)
+ getSocket() : int
+ closeTcpSocket() : void
SocketFetching
- messengers : Lyst<ChcMessenger>
+ SocketFetching()
+ SocketFetching()
+ addMessenger(ChcMessenger messenger) : void
+ run() : void
+ run() : void
+ sleep() : void
ClcCommand
+ type : String
+ ClcCommand(String type)
NeutralEvent
+ NEW_USER : String
+ pseudo : String
+ pseudo : String
+ NeutralEvent(int userId, String pseudo)
LoginEvent
+ LOGIN_SUCCESS : String
+ LOGIN_FAILED : String
+ LoginEvent(String event)
MessageEvent
+ MESSAGE_EVENT : String
+ MessageEvent()
RemoveUserEvent
+ REMOVE_USER : String
+ pseudo : String
+ id : int
+ RemoveUserEvent(String pseudo, int id)
TypingEvent
+ TYPING : String
+ TYPING_END : String
+ sharedId : String
+ isTyping : int
+ TypingEvent(int userId , String sharedId, boolean isTyping)
ChcNetworkMessage
+ MAXT : String
+ ENCRYPTED_MSG : String
+ DISCOVER_REQUEST : String
+ type : String
+ ChcNetworkMessage(String type)
DiscoverResponseMessage
+ DISCOVER_RESPONSE : String
+ count : int
+ avatar : byte[]
+ pseudo : String
+ id : int
+ DiscoverResponseMessage(String pseudo, int id, byte[] avatar, int count)
SharedMessage
+ SHARED_ID : String
+ sharedId : String
+ pseudo : String
+ id : int
+ SharedMessage(String sharedId, int id, String pseudo)
TypingMessage
+ TYPING_START : String
+ TYPING_END : String
+ sharedId : String
+ isTyping : int
+ TypingMessage(String type, int userId, String sharedId)
TextMessage
+ TEXT_NETWORK_MESSAGE : String
+ sharedId : String
+ message : String
+ userId : int
+ TextMessage(int userId, String sharedId, String message)
SessionMessage
+ LOGIN : String
+ LOGOUT : String
+ PSEUDO : String
+ id : int[]
+ ok : int
+ SessionMessage(String type)
+ SessionMessage(String type, String pseudo, int id, byte[] mp)
+ SessionMessage(String type, String pseudo, int id)