

Kevin MAUGE : kmauge@etud.u-pem.fr  
Pape NDIAYE : pndiaye@etud.u-pem.fr

Name : Matou, a Chat service

#### Description:

The purpose of Matou project is to realize a discussion and file exchanger service.

The clients are connected to a primary server. Each connected client is identified by a unique pseudo. The server must guarantee that two client will not have the same pseudo. After establishing the connection, the client can:

- \* Send message to all connected clients
- \* Send privates messages or files to a particular client

#### Requirements:

- \* All private communication are made by a direct connection between the two clients. The server will only be sollicited for the purpose of exchanging the address of those two clients.

- \* Client will be managed to send a request for private messaging. And this client can't accept or refuse any private request.

- \* Exchanging file should not block the messaging and vice versa.

#### Specifications:

- \* The numbers are in Big Endian format
- \* String are encoded into UTF-8 charset
- \* The transport protocol used is TCP (with limit size specified)
- \* The max size of an encoded pseudo is 32 bytes wihtout some special characters (\n, \r ...)
- \* The max size of a message is 512 bytes
- \* The size of a chunk of file is 512 bytes

#### Acronym formats:

- \* C-S : Client to Server
- \* S-C : Server to Client
- \* S-A : Server to All Client
- \* C-C : Client to Client

## Interactions:

### Request for public connection

[C-S] Request connection to Server

The client requests a connection to the server with a specified pseudo

```
+-----+-----+-----+
| int  |    int    | String |
+-----+-----+-----+
| Type | SizePseudo | Pseudo |
+-----+-----+-----+
```

[S-C] Validation of the connection

The server answers with a boolean which will indicate whether the pseudo is valid or not. If the pseudo is valid the connection is established.

```
+-----+-----+
| int  |    byte    |
+-----+-----+
| Type | Acceptation |
+-----+-----+
```

[S-A] Notification of the new connection

The server notifies all clients that a new one is connected.

```
+-----+-----+
| int  |    int     | String |
+-----+-----+
| Type | SizePseudo | Pseudo |
+-----+-----+
```

### Sending message to the public domain

[C-S] Sending message

The client sends his public message to the server.

```
+-----+-----+-----+
| int  |    int     | String |
+-----+-----+-----+
| Type | SizeMessage | Message |
+-----+-----+-----+
```

[S-A] Retransmission of the message

The server forwards the public message to all connected clients.

```
+-----+-----+-----+-----+-----+
| int  |    int     | String |    int     | String |
+-----+-----+-----+-----+-----+
| Type | SizePseudo | Pseudo | SizeMessage | Message |
+-----+-----+-----+-----+-----+
```

## Requesting private connection

[C-S] Requesting connection between the source(Src) and the destination(Dst)

The client Src notifies the server that he wants to be connected with the client Dst.

int	int	String
Type	SizePseudoDst	PseudoDst

[S-C] Requesting acceptance of the connection from Dst

The server notifies the client Dst that he received a connection request from client Src.

int	int	String
Type	SizePseudoSrc	PseudoSrc

[C-S] Confirm the connection request to Src

The client Dst notifies the server whether he accepts or refuses the private connection request of client Src.

int	int	String	byte
Type	SizePseudoSrc	PseudoSrc	Accept

[S-C] Notification of the private

The server forwards the answer of client Dst to client Src.

int	int	String	byte
Type	SizePseudoDst	PseudoDst	Accept

[S-C] Connection between the source(Src) and the destination(Dst)

The server sends the private connection data for the private connection of client Dst to client Src.

int	int	String	InetSocketAddress
Type	SizePseudoDst	PseudoDst	AddressDst

[S-C] Connection between the source(Src) and the destination(Dst)

The server sends the private connection data for the private connection of client Src to client Dst.

int	int	String	InetSocketAddress	int
Type	SizePseudoSrc	PseudoSrc	AddressSrc	PortSrc

#### Sending private message

[C-C] Sending private message between Src and Dst  
The client Src sends a message to client Dst.

```
+-----+-----+-----+
| int  |      int      | String |
+-----+-----+-----+
| Type | SizeMessage | Message |
+-----+-----+-----+
```

#### Sending file in private

[C-C] Sending file between Src and Dst  
The client Src sends a file to client Dst. The file is segmented into chunk with a fixed size (only the last part will eventually be smaller).

```
+-----+-----+-----+
| int  |      long      |      bytes  | bytes |
+-----+-----+-----+
| Type | SizeFile  | FileChunk | ...  |
+-----+-----+-----+
```

#### Private sign out

[C-C] Sign out  
The client notifies his interlocutor that he wants to terminate the private connection.

```
+-----+
| int  |
+-----+
| Type |
+-----+
```

#### Public sign out

[C-S] Sign out  
The client notifies the server that he wants to terminate the connection.

```
+-----+
| int  |
+-----+
| Type |
+-----+
```

#### [S-A] Sign out notification

The server notifies all clients that a client has left the chat room.

```
+-----+-----+-----+
| int  |      int      | String |
+-----+-----+-----+
| Type | SizePseudo | Pseudo |
+-----+-----+-----+
```