RFC A Dog Is Not A Tchat Protocol

# Introduction

The objective of A Dog Is Not A Tchat (ADINAT) protocol is to send and receive text messages and files. This protocol depends on the Transmission Control Protocol (TCP) in order for the server to communicate with the clients and vice-versa. This protocol receives connection requests from clients, then the clients must assign themselves usernames.

**Index**

[Introduction 1](#_7mu2tnjrjifh)

[Basic Operation 2](#_i1u9y9bpj5tb)

[I. Commands from Client to Server 3](#_utu3vgacg82z)

[- help 3](#_k506xbe42qzs)

[- msg <message> 3](#_9ify61mhmtk)

[- msgpv <username> <message> 4](#_ia2dzi8kr5qg)

[- exit 5](#_bqp373obguk)

[- afk 5](#_cgzckygadph3)

[- btk 5](#_wr3wjjets7xe)

[- users 6](#_vto1lcbuzk9k)

[- rename <username> 6](#_wa8hcl5eu8mw)

[- ping <username> 7](#_17pj0ypmxdqd)

[- channel <username> 7](#_juvtm8yy8jl)

[- acceptchannel <username> 8](#_itvgm4jpsdtn)

[- declinechannel <username> 8](#_oa3sa1o8mdar)

[- sharefile <username> <file path> <port> 9](#_ldm4llj9363b)

[- acceptfile <username> <namefile> 10](#_7oid266byogi)

[- declinefile <username> <namefile> 11](#_9dbhpwtlxolb)

[II. Responses from server to clients 12](#_vturk8x4buwy)

[- signupFromSrv 12](#_nklvyd9711ak)

[- msgFromSrv 12](#_u7ic35yn18a)

[- msgpvFromSrv 12](#_2tyklessp65z)

[- exitedFromSrv 12](#_thzb4latlw79)

[- afkFromSrv 13](#_ffrodhzfy0m9)

[- btkFromSrv 13](#_1tvoil1vq4yt)

[- usersFromSrv 13](#_q9fgw3z5avf6)

[- renameFromSrv 13](#_m9932jjeclzw)

[- pingFromSrv <usernameSource> 14](#_7bccidsnaxqu)

[- channelFromSrv 14](#_whmywi5ru02e)

[- acceptedchannelFromSrv 14](#_44hlisjars0a)

[- declinedchannelFromSrv 14](#_fn6cat131icm)

[- sharefileFromSrv 15](#_oj4fxqxummlx)

[- acceptedfileFromSrv 15](#_njmeza8in9z)

[- declinedfileFromSrv 15](#_a56xde7iaxcq)

[Summarization 16](#_7e9tzqd4mw1r)

[Response Codes 17](#_y0vplke2790h)

## Note

This RFC consists of 2 parts.

The first part represents the commands that a client may call. Each command lists the requirements for the client in order for the command to succeed. In this part, the server’s response will only be shown as a return code sent to the client who sent the command.

The second part consists of the server’s responses to one or multiple clients once the client who issued a command is successful.

## Basic introduction

The basic use of using a command is by using the command name itself, without spaces in between. In order to use a command that includes a number of arguments, the same writing is used again followed by a single space and then the argument.

The character “|” is used for separation when a command transits between a client and the server. It is not accepted in any username, message or parameter.

In this section you will find a listing of specifications of the syntax used to create this RFC, to facilitate the comprehension of the document.

The syntax <> refers to a parameter, meaning that the content from an opening “<” until a closing “>” should be replaced with a real variable. For instance, an example in this RFC would be “StoCB: channelFromSrv|<client A’s username>”. In a real life example, the variable “<client A’s username>” should be replaced by the actual client A’s username. Considering client A’s username is “toto” the server’s response will be “StoCB:channelFromSrv|toto”

Concerning the examples, the acronyms C, CA, CB, AC, S, CtoS, StoC, StoCA, StoCB, StoAC stand for :

* C : Client
* CA : Client A
* CB : Client B
* AC : All connected clients
* S : Server
* CtoS : Command sent to a server from a Client
* StoC : Server’s response sent to the client who issued the command
* StoCA : Server’s response sent to Client A
* StoCB : Server’s response sent to Client B
* StoAC : Server’s response to all connected clients

# Commands from Client to Server

## help

* + Sends to the client all existing commands in a list under the form **[afk, btk, exit, users…..]**

Exemple :

CtoS: help

StoC: 200

**signup <username>**

* The client that sent the command tries to join the chatroom by giving an username in the parameter.
* The username must be comprised of only letter characters (e.g. a, b, c, d etc.), meaning no special characters (e.g. ^,$.& etc.), spaces, nor numbers (e.g. 1, 12, 23.34, 435,000) will be allowed. If the username isn’t valid, the server will issue the response “426”.
* The username must be unique in regards to the usernames of other clients. If the username is already taken by another client, the server will issue the response “425”
* This command can be issued only upon connecting to the server. Once the user “signed up”, he can no longer issue the command. If the user is already connected, the server will issue the response “417”.
* This command must have a single parameter, in the event that no parameter is given or that multiple parameters are given, the server will issue the response “403”.
* The server will send the response “200” if all the constraints are respected.
* Once the user is connected, his state changes to btk state.

## msg <message>

* + The client that sent the command sends the message given in the parameter to the other connected clients.
  + The client must be logged in, otherwise the server will issue the response “418”
  + The client must be in the ‘afk’ state, otherwise the server will issue the response “430”
  + This command must have a single parameter, in the event that no parameter is given or that multiple parameters are given, the server will issue the response “403”

Exemple 1 (no errors):

CtoS : msg <message>

StoC : 200

Exemple 2 (client is not logged in):

CtoS : msg <message>

StoC : 418

## msgpv <username> <message>

* + The client that sent the command sends the message given in the second parameter to the other connected client given in the first parameter.
  + The private message will be sent only if there already exists a private channel between the concerned clients. In order to request a private channel, the client must issue the command “channel <username>”. If no channel is established, the server will issue the response “421”.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
  + If the username specified in the parameter doesn’t exist, the server will issue the response “402”.
  + The client issuing the command must not enter his own username, otherwise, the server will issue the command “407”.
  + This command must have two parameters, in the event that another number of parameters are given, the server will issue the response “403”.
  + If all constraints are respected, the server will issue the response “200”.

Exemple 1 (no errors):

C1toS : msgpv <usernameC2> <message>

StoC1 : 200

Exemple 2 (errors):

C1toS : msgpv <username2> <message>

StoC1 : 401

Exemple 3 (username doesn’t exist)

C1toS : msgpv <username2> <message>

StoC1 : 402

Example 4 (no parameter is given OR multiple parameters are given):

C1toS : msgpv <usernameC2> <message><something>

StoC1 : 403

Example 5 (Example with an accept) :

C1toS : channel <username2>

StoC1 : 200

C2toS: acceptchannel <username1>

StoC2 : 200

C1toS : msgpv <username2>

StoC1 : 200

Example 6 (Private channel does not exist) :

CtoS : msgpv <username2> <message>

StoC : 421

## exit

* + The client who sent the command terminates the connection between himself and the server
  + The server will notify all connected clients that the client has ended his connection
  + Client’s session is closed on the server
  + All created channels for the client that disconnected become closed

Example 1 (client left the server):

CtoS : exit

StoC : 200

Example 2 (error):

CtoS : exit

StoC : 401

## afk

* + The client that sent the command can’t send any commands to the server during afk time except the “btk” command.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + If the client is already “afk” and it tries to send the “afk” command, the server will issue the response “415”
  + If all constraints are respected, the server will issue the response “200”.

Exemple 1 (client is already afk) :

CtoS: afk

StoC: 415

Exemple 2 (client is not afk):

CtoS: afk

StoC: 200

## btk

* + The client that sent the command that was in the “afk” state becomes active again allowing him to issue commands to the server.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + If the client who sent the command is in the ‘btk’ state, the server will issue the response “416”.
  + If all constraints are respected, the server will issue the response “200”.

Exemple 1 (client is already btk) :

CtoS: btk

StoC: 416

Exemple 2 (client is not afk) :

CtoS: btk

StoC: 200

## users

* + Notifies the client that sent the command which clients are connected to the server by sending a list of the clients’ usernames.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
* If all constraints are respected, the server will issue the response “200”.

Example 1 (at least 1 user is online):

CtoS : users

StoC : 200

## rename <username>

* + The client that sent the command is able to change his username.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
  + In the event that the username is taken by another client, the server will issue the response “425”.
  + This command must have a single parameter, in the event that no parameter is given or that multiple parameters are given, the server will issue the response “403”.
  + The username must not contain any numbers or special characters, otherwise the server will issue the response “426”.
  + If all constraints are respected, the server will issue the response “200”.

Example 1 (username already exists):

CtoS : rename hortencia

StoC : 425

Example 2 (username contains numbers or special characters):

CtoS : rename hortencia24

StoC : *426*

*Example 3 (no parameter is given OR multiple parameters are given):*

*C*toS *: rename*

*S*toC *: 403*

## ping <username>

* + The client that sent the command pings the client given in the parameter.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
  + If the username specified in the parameter doesn’t exist, the server will issue the response “402”.
  + The client issuing the command must not enter his own username, otherwise, the server will issue the command “407”.
  + This command must have a single parameter, in the event that no parameter is given or that multiple parameters are given, the server will issue the response “403”.
  + If all constraints are respected, the server will issue the response “200”.

Example 1 (username doesn’t exist):

CtoS : ping username

StoC : 402

Example 2 (username exists) :

CtoS: ping username

StoC : 200

## channel **<username>**

* + The client that sent the command requests a private channel between him and the client whose username is specified in the parameter.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
  + If the username specified in the parameter doesn’t exist, the server will issue the response “402”.
  + This command must have a single parameter, in the event that no parameter is given or that multiple parameters are given, the server will issue the response “403”.
  + The client issuing the command must not enter his own username, otherwise, the server will issue the command “407”.
  + In the case that a channel already exists between the client who sent the command and the target client given in the parameter, the server will issue the response “404”.
  + In the event that the client already issued this command to the client given in the parameter, if he issues it again, the server will issue the response “441” until the targeted client accepts or declines the request.
  + If all constraints are respected, the server will issue the response “200”.

Example 1 (username doesn’t exist) :

CtoS : channel <username2>

StoC : 402

Example 2 (username exists):

C1toS : channel <username2>

StoC : 200

Example 3 (Private channel already exists) :

CtoS : channel <username2>

StoC : 404

## acceptchannel **<username>**

* + This command is bound to the “channel” command.
  + The client that was in the parameter of the “channel” command can send this command in order to accept the request.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
  + If the username specified in the parameter doesn’t exist, the server will issue the response “402”.
  + The client issuing this command must enter the username of the client who made the request, otherwise the server will issue the response “444”.
  + The client issuing the command must not enter his own username, otherwise, the server will issue the command “407”.
  + The client issuing this command can only do it if he received a private channel request, otherwise the server will issue the response “440”.
  + This command must have a single parameter, in the event that no parameter is given or that multiple parameters are given, the server will issue the response “403”.
  + If all constraints are respected, the server will issue the response “200”.

Example 1 (username and request exist) :

CtoS : acceptchannel <username2>

StoC1 : 200

Example 2 (username doesn’t exist) :

CtoS : acceptchannel <username2>

StoC : 402

## declinechannel <username>

* + This command is bound to the “channel” command.
  + The client that was in the parameter of the “channel” command can send this command in order to decline the request.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
  + If the username specified in the parameter doesn’t exist, the server will issue the response “402”.
  + The client issuing this command must enter the username of the client who made the request, otherwise the server will issue the response “444”.
  + The client issuing the command must not enter his own username, otherwise, the server will issue the command “407”.
  + The client issuing this command can only do it if he received a private channel request, otherwise the server will issue the response “440”.
  + This command must have a single parameter, in the event that no parameter is given or that multiple parameters are given, the server will issue the response “403”.
  + If all constraints are respected, the server will issue the response “200”.

Example 1 (username and request exist):

C : declinechannel <username2>

StoC : 200

Example 2 (username doesn’t exist):

CtoS : declinechannel <username2>

StoC : 402

## sharefile <username> <file path> <port>

* The client that sent the command requests the transfer of a file which he has on his local disk to another client. This command requires the path to the file and the port number to propose for the other client in order to make a P2P (peer to peer) transfer.
* The client issuing the command must be logged in, otherwise the server will issue the response “418”.
* The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
* If the username specified in the parameter doesn’t exist, the server will issue the response “402”.
* The client issuing the command must enter a valid file path to the file he wishes to share, otherwise the server will issue the response “405”
* The client issuing the command must give a port number that is valid and open, otherwise the server will issue the response ”446”.
* The client issuing the command must not enter his own username, otherwise, the server will issue the command “407”.
* This command must have three parameters, in the event that less than three parameters are given or that more than three parameters are given, the server will issue the response “403”.
* In the event that the client already issued this command to the client with the same file given in the parameters, if he issues it again, the server will issue the response “442” until the targeted client accepts or declines the request.
* If all constraints are respected, the server will issue the response “200”.

Example 1 (username does not exist):

CtoS : sharefile <username2> <namefile> <port>

StoC : 402

Example 2 (path file does not exist):

CtoS : sharefile <username2> <namefile> <port>

StoC : 405

Example 3 (username and pathfile exist):

CtoS : sharefile <username2> <namefile> <port>

StoC : 200

Example 4(no parameter is given OR too many parameters are given):

CtoS : sharefile <port> <namefile>

StoC : 403

## acceptfile <username> <namefile>

* + This command is bound to the “sharefile” command.
  + The client that was in the parameter of the “sharefile” command can send this command in order to accept the request and initiate the transfer.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
  + If the username specified in the parameter doesn’t exist, the server will issue the response “402”.
  + The client issuing the command must not enter his own username, otherwise, the server will issue the command “407”.
  + The client issuing this command must enter the username of the client who made the request, otherwise the server will issue the response “445”.
  + This command must have two parameters, in the event that no parameter is given or that more than two parameters are given, the server will issue the response “403”.
  + The client issuing this command can only do it if he received a transfer file request, otherwise the server will issue the response “443”.
  + If the name file specified in the parameter doesn’t exist (meaning it is not the same name as the name of the file given by the transmitter), the server will issue the response “406”.
  + If all constraints are respected, the server will issue the response “200”.

Example 1 (username and namefile exists) :

CtoS : acceptfile <username2> <namefile>

StoC : 200

Example 2 (username doesn’t exist) :

CtoS : acceptfile <username2> <namefile>

StoC : 402

Example 3 (namefile doesn’t exist) :

CtoS : acceptfile <username2> <namefile>

StoC : 405

Example 4 (no parameter is given OR too many parameters are given):

CtoS : acceptfile <username2> <namefile> <param3>

StoC : 403

## declinefile <username> <namefile>

* + This command is bound to the “sharefile” command.
  + The client that was in the parameter of the “sharefile” command can send this command in order to decline the request made by the other client.
  + The client issuing the command must be logged in, otherwise the server will issue the response “418”.
  + The client issuing the command must be in the ‘btk’ state, otherwise the server will issue the response “430”.
  + If the username specified in the parameter doesn’t exist, the server will issue the response “402”.
  + The client issuing the command must not enter his own username, otherwise, the server will issue the command “407”.
  + The client issuing this command must enter the username of the client who made the request, otherwise the server will issue the response “445”.
  + The client issuing this command can only do it if he received a transfer file request, otherwise the server will issue the response “443”.
  + This command must have two parameters, in the event that no parameter is given or that more than two parameters are given, the server will issue the response “403”.
  + If the name file specified in the parameter doesn’t exist (meaning it is not the same name as the name of the file given by the transmitter), the server will issue the response “406”.
  + If all constraints are respected, the server will issue the response “200”.

Example 1 (username and file exist):

CtoS : declinefile <username2> <namefile>

StoC : 200

Example 2 (username doesn’t exist):

CtoS : declinefile <username2> <namefile>

StoC : 402

Example 3 (namefile doesn’t exist):

CtoS : declinefile <username2> <namefile>

StoC : 405

# Responses from server to clients

The following list contains a list of responses issued by the server to either client A (the client who issued the command), client B (the client who was given in the parameter when client A issued the command) or all other clients connected to the chatroom. All the responses issued by the server on this list are made ONLY if client A met the requirements to issue the specific command listed in the first part of this RFC (meaning client A received the response “200” for the specific command).

## signupFromSrv

* + If clientA met all the requirements upon issuing the ‘signup’ command, the server will issue the response “signupFromSrv|<client A’s username>” to all connected clients.

Exemple :

CAtoS : signup Dan

StoAC: signupFromSrv|Dan

## msgFromSrv

* + If client A met all the requirements upon issuing the ‘msg’ command, the server will issue the response “msgFromSrv|<clientA’s username>|<message>” to all connected clients.

Exemple :

CAtoS : msg hello everyone, how’s it going ?

StoC : msgFromSrv|<client A’s username>|hello everyone, how’s it going ?

## 

## msgpvFromSrv

* If client A met all the requirements upon issuing the ‘msgpv’ command, the server will issue the response “msgpvFromSrv|<clientA’s username>|<message>” to client B.

Exemple:

CAtoS : msgpv Louis hey Louis, what’s new ?

StoCB : msgpvFromSrv|<client A’s username>|hey Louis, what’s new ?

## exitedFromSrv

* If client A met all the requirements upon issuing the ‘exit’ command, the server will issue the response “exitedFromSrv|<clientA’s username>” to all connected clients.

Example:

CAtoS : exit

StoAC : exitedFromSrv|<client A’s username>

## afkFromSrv

* + If client A met all the requirements upon issuing the ‘afk’ command, the server will issue the response “afkFromSrv|<clientA’s username>” to all connected clients.

Exemple:

CAtoS : afk

StoAC: afkFromSrv|<client A’s username>

## btkFromSrv

* If client A met all the requirements upon issuing the ‘btk’ command, the server will issue the response “btkFromSrv|<clientA’s username>” to all connected clients.

Exemple :

CAtoS : btk

StoC: btkFromSrv|<client A’s username>

## usersFromSrv

* + If client A met all the requirements upon issuing the ‘btk’ command, the server will issue the response “usersFromSrv|[username1, username2, …. , usernameK]” to client A.
  + The response consists of a list of connected client’s usernames with K being the number of users connected to the server.
  + The response must be exactly under this format starting with an opening bracket, then name of an user followed by comma and a space and so on, finishing up with a closing bracket.
* The response also contains the username of the client A.

Example :

CAtoS : users

StoCA : users|[username1, username2, username3]

## renameFromSrv

* If client A met all the requirements upon issuing the ‘rename’ command, the server will issue the response “renameFromSrv|<client A’s old username>|<client A’s new username>” to all connected clients.

Example :

CAtoS : rename Marvin

StoAC : renameFromSrv|<client A’s old username>|Marvin

## pingFromSrv <usernameSource>

* If client A met all the requirements upon issuing the ‘ping’ command, the server will issue the response “pingFromSrv|<client A’s username>” to client B.

Example :

CAtoS : ping Louis

StoCB : pingFromSrv|<client A’s username>

## channelFromSrv

* If client A met all the requirements upon issuing the ‘channel’ command, the server will issue the response “channelFromSrv|<client A’s username>” to client B.

Example :

CAtoS : channel Louis

StoCB : channelFromSrv|<client A’s username>

## acceptedchannelFromSrv

* If client A met all the requirements upon issuing the ‘acceptchannel’ command, the server will issue the response “acceptedchannelFromSrv|<client A’s username>” to client B.

Example :

CAtoS : acceptchannel Marvin

StoCB : acceptedchannelFromSrv|<client A’s username>

## declinedchannelFromSrv

* If client A met all the requirements upon issuing the ‘declinechannel’ command, the server will issue the response “declinedchannelFromSrv|<client A’s username>” to client B.

Example :

CAtoS : declinechannel Marvin

StoC : declinedchannelFromSrv|<client A’s username>

## sharefileFromSrv

* If client A met all the requirements upon issuing the ‘sharefile’ command, the server will issue the response “sharefileFromSrv|<client A’s username>|<name file>|<size file>|<client A’s host address>|<port number>” to client B.

Example :

CAtoS : sharefile Marvin ./temp/school/FunnyImage.png 56789

StoCB: sharefileFromSrv|<client A’s username> | FunnyImage.png | 1.3 MB | 192.102.0.1 | 56789

## acceptedfileFromSrv

* If client A met all the requirements upon issuing the ‘acceptfile’ command, the server will issue the response “acceptfileFromSrv|<client A’s username>|<name file>” to client B.

Example :

CAtoS : acceptfile Louis FunnyImage.png

StoCB : acceptedfileFromSrv|<client A’s username>|FunnyImage.png

## declinedfileFromSrv

* If client A met all the requirements upon issuing the ‘declinefile’ command, the server will issue the response “declinedfileFromSrv|<client A’s username>|<name file>” to client B.

Example :

CAtoS : declinefile Louis FunnyImage.png

StoC : declinedfileFromSrv|<client A’s username>|FunnyImage.png

## 

# 

# Summarization

The server responses can be categorized into 4 sets.

First set of responses consisting of return codes only issued to client A.

Second set of responses are only issued to client A once the response is “200”.

Third set of responses are issued to client B (the client passed in the parameter of the command by client A).

The fourth set of responses are issued to all connected clients.

First set:

| **Command (from client A)** | **Response (to client A)** |
| --- | --- |
| signup | 403 OR 417 OR 425 OR 200 |
| msg | 403 OR 418 OR 430 OR 200 |
| msgpv | 403 OR 418 OR 430 OR 402 OR 407 OR 421 OR 200 |
| exit | 403 OR 200 |
| afk | 403 OR 418 OR 415 OR 200 |
| btk | 403 OR 418 OR 416 OR 200 |
| users | 403 OR 418 OR 430 OR 200 |
| rename | 403 OR 418 OR 430 OR 425 OR 426 OR 200 |
| ping | 403 OR 418 OR 430 OR 402 OR 407 OR 200 |
| channel | 403 OR 418 OR 430 OR 402 OR 407 OR 404 OR 441 OR 200 |
| acceptchannel | 403 OR 418 OR 430 OR 402 OR 407 OR 404 OR 440 OR 444 OR 200 |
| declinechannel | 403 OR 418 OR 430 OR 402 OR 407 OR 404 OR 440 OR 444 OR 200 |
| sharefile | 403 OR 418 OR 430 OR 402 OR 405 OR 407 OR 446 OR 442 OR 200 |
| acceptfile | 403 OR 418 OR 430 OR 402 OR 443 OR 407 OR 445 OR 406 OR 200 |
| declinefile | 403 OR 418 OR 430 OR 402 OR 443 OR 445 OR 406 OR 200 |

Second set :

| **Command (from client A)** | **Response (to client A)** |
| --- | --- |
| users | usersFromSrv|[username1, username2, …] |

Third set :

| **Command (from client A)** | **Response (to client B)** |
| --- | --- |
| msgpv usernameB text | msgpvFromSrv|usernameA|text |
| ping usernameB | pingFromSrv|usernameA |
| channel usernameB | channelFromSrv|usernameA |
| acceptchannel username | acceptchannelFromSrv|usernameA |
| declinechannel username | declinechannelFromSrv|usernameA |
| sharefile username filepath | sharefileFromSrv|usernameA|filename |
| acceptfile username filename | acceptfileFromSrv|usernameA|filename |
| declinefile username name | declinefileFromSrv|usernameA|filename |

Forth set :

| **Command (from client A)** | **Response (to all clients)** |
| --- | --- |
| signup usernameA | signupFromSrv|usernameA |
| afk | afkFromSrv|usernameA |
| btk | btkFromSrv|usernameA |
| msg text | msgFromSrv|usernameA|text |
| exit | exitedFromSrv|usernameA |
| rename newName | renameFromSrv|oldUsernameA|newUsernameA |

# Response Codes

This section summarizes the set of codes that must be returned by a server configured with an ADINAT protocol and the reason why theu may be returned:

| N° : | Reason: |
| --- | --- |
| 200 | Success |
| 400 | The command doesn’t exist probably due to a typing error |
| 401 | Message error |
| 402 | Username does not exist |
| 403 | Wrong numbers of parameters |
| 404 | Private channel already exists |
| 405 | File name does not exist |
| 406 | The name of the file given by the user isn’t corresponding |
| 407 | You are not authorized to send messages to yourself |
| 415 | Already afk |
| 416 | Already btk |
| 417 | User already logged in |
| 418 | User must be logged in |
| 421 | Private channel does not exist |
| 425 | Username already taken |
| 426 | Username must not contain special characters or numbers |
| 430 | User is afk |
| 440 | You have no pending private channel requests |
| 441 | You have already sent a private channel request to that user |
| 442 | You have already sent a file transfer request to that user |
| 443 | You have no pending share file requests |
| 444 | You have no pending private channel request from that user |
| 445 | You have no pending share file requests from that user |
| 446 | Not a valid port |
| 500 | Internal server error |

**Brouillon :**

sharefile : envoie la commande sharefile avec le pseudo du destinataire et le chemin absolue du fichier à envoyer sur le channel privé

La taille du fichier sera affiché au destinataire avant qu’il accepte de recevoir le fichier.

(A afficher ou non car c’est de l’implémentation) Dans le cas où le channel n’est pas encore créé : la commande channel sera d’abord exécutée. Le destinataire exécutera la commande accept ou decline selon sa volonté.)

cmd7 : Demande de création de chanel

cmd8 :

- accept : crée un channel entre les deux personnes

- decline : pas de chanel créé, renvoie à celui qui demande que la demande a été refusé

Status error :

200 -> ok

401 -> quand un utilisateur n’a pas de username ou n’as pas les droits

403 -> le serveur refuse d’envoyer les informations suite à un manque d’information du client (username)

409 -> afk alors que déjà afk ou btk alors que déjà btk

404 -> pas d’utilisateur trouvé

400 -> écrire sharefile au lieu de sharefile

Commandes :

msg : envoie un message dans le channel général

msgpv <username> : envoie un message dans le channel privé avec l’user en question

exit : Ferme la connexion entre le serveur et le client, envoie l’information de fermeture de la connexion au serveur, fermeture du socket client.

afk : Le client reste connecté et reçoit les notifs. La commande d’envoie de message est bloquée.

btk : Le client qui était afk redevient actif

users : liste les utilisateurs connectés

rename : Permet à l’utilisateur de changer son pseudo

ping : Un client ping un pseudo

channel :

help : affiche la liste de toutes les commandes avec leur nomenclature à la personne qui vient de faire cette commande.

Server’s response to the clients concerned by command :

| **Commands** | **Response** |
| --- | --- |
| msg text | msg|username|text |
| msgpv receiver texte | msgpv|usernameSender|usernameReceiver|text |
| exit | exit|username |
| afk | afk|username |
| btk | btk|username |
| rename newName | rename|oldUsername|newUsername |
| ping username | ping|username |
| channel username | channel|username |
| acceptchannel username | acceptchannel|username |
| declinechannel username | declinechannel|username |
| sharefile username filename | sharefile|username|namefile |
| acceptfile username namefile | acceptfile|username|namefile |
| declinefile username namefile | acceptfile|username|namefile |

Server’s response to the client who sent the command:

| **Commands** | **Response** |
| --- | --- |
| msg text | codeError|msg||text |
| msgpv receiver texte | codeError|msgpv|usernameReceiver|text |
| help | codeError|help|text |
| exit | codeError|exit|text |
| afk | codeError|afk|text |
| btk | codeError|btk|text |
| users | codeError|users|text |
| rename newName | codeError|rename|newUsername|text |
| ping username | codeError|ping|username|text |
| channel username | codeError|channel|username|text |
| acceptchannel username | codeError|acceptchannel|username|text |
| declinechannel username | codeError|declinechannel|username|text |
| sharefile username filename | codeError|sharefile|username|namefile|text |
| acceptfile username namefile | codeError|acceptfile|username|namefile|text |
| declinefile username namefile | codeError|acceptfile|username|namefile|text |