“Posting Ads” Protocol

Note: Circulating messages are indicated in square brackets, the brackets not part of the message. For error messages, the idea is to have "one code per type of error ". All messages have a maximum of 1024 bytes.

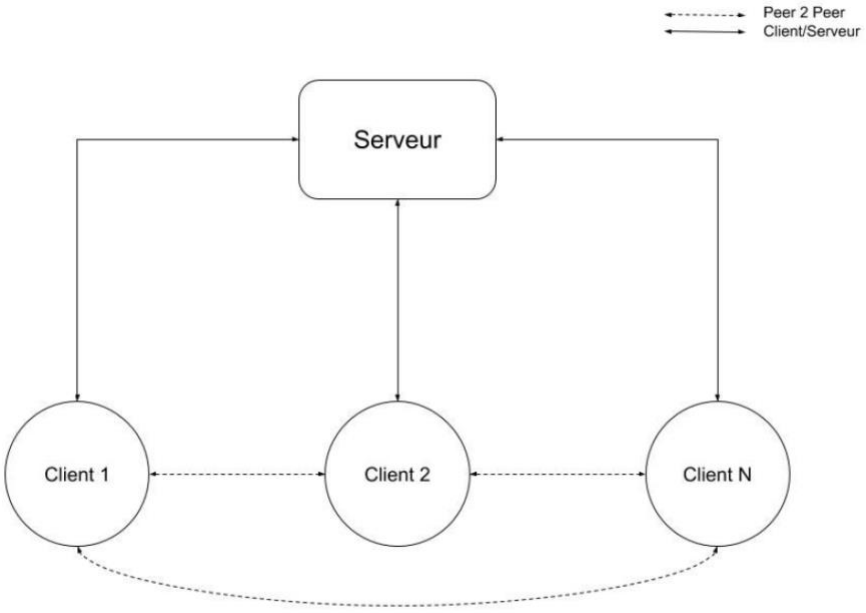
Client:

The characteristics of a customer are as follows:
- a login
- a password
- an IP address and a TCP port so that other customers can contact him
- a boolean to indicate if it is connected or not
- a ServerSocket to communicate with other clients

Server:

The characteristics of a server are as follows:
- a list of clients
- TCP port to accommodate customers (2345)
- a HashMap (yes I love hashmaps) <Key = Ad number, Value = Associate client>

Architecture



Client-server interaction

Register:

1) The client sends: [REGI id mdp +++]
2) The server responds: - [OKOK +++] if everything went well –
[KOKO +++ code] if there is an error (codes 0 or 1) and close the connection

To log in:

1) The client sends: [CONE id mdp port +++]
2) The server responds: - [OKOK +++] if everything went well - [KOKO +++ code] if there is an error (codes 1, 2 or 3) and close the connection

Sign out:

1) The customer sends: [DECO +++]
2) The server responds [GBYE +++] and closes the connection

Post an ad:

1) The client sends: [POST code descriptive price +++] (code is an integer ranging from 1 to 6 indicating the category of the ad - see below -, price is in _._ format, description is a string describing the ad)

Categories of ads (1 to 6)
1: Car 2: Motorcycle
3: Music
4: Home appliance
5: Phones
6: other.

2) The server responds: - [OKOK id +++] if everything went well (id being the unique id of the ad generated by the server) - [KOKO +++ code] if there is an error (codes 1 or 4)

Delete an ad:

1) The client sends: [SUPR idAd +++]
2) The server responds:
- [OKOK +++] if everything went well
- [KOKO +++ code] if there is an error (codes 1, 4, 5 or 6)

List all the ads

1) The client sends: [LIST +++]
2) The server responds:
- [NBAN nb +++] nb is the number of ads
- [ANNO id code price description +++] for each ad

Retrieve ad information

1) The client sends: [INTR idAds +++]
2) The server responds:
- [CONT ip port +++] with ip address and port the information of the owner of
the announcement
- [NCON +++] if the owner of the ad is not connected
- [KOKO +++ code] if there is an error (codes 1 or 5)

Juste a suggestion: (idea: two possibilities for id / port:
1) Store at the customer, for each requested ad, the ip and the port of the owner.
Thus, the customer will only have to say in input "I want to buy that" and the program will look for
ip itself and port (= lazy client)
Example via two HashMap (yes again!): <Key = NumAnnonce, Value = ip> and
<Key = NumAd, value = port>
OR a numAnnonce-Infos couple but that implies to split the string afterwards ...
2) Let the customer indicate numAnnonce, id and port when buying (= smart client)

Interaction between clients

Communication between clients will be done using TCP.

Initialize the exchange and check availability

1) The client sends: [DISP idAds+++] to the owner of the ad
2) The customer owner answers:
- [OKOK +++] if the ad is available
- [KOKO +++ code] if there is an error (codes 1, or 5)

Send a message

1) The client sends: [MSSG text+++] to the owner of the ad
2) The client owner answers: [MSSG text+++]

Buy the product

1) Customer sends: [ACHA+++] to the owner of the ad

2) The customer owner answers:
- [OKOK +++] if he agrees
- [KOKO code+++] if there is an error (codes 1 or 8)

The connection is closed after the purchase

End the exchange

1) The customer sends: [GBYE+++] to the owner of the ad
2) The owning customer responds: [GBYE+++]

The connection is closed after sending this message

Reference of error codes

Code 0: ID already exists
Code 1: Wrong number of arguments
Code 2: Incorrect password
Code 3: The user does not exist
Code 4: Incorrect format of arguments (for example, out-of-format price _._)
Code 5: The ad does not exist
Code 6: The customer does not own the ad
Code 7: Command error (for example, SUP instead of SUPR)
Code 8: Client refuses to sell the product.