WARNING:

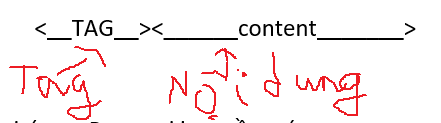
Clone code mới trên git Sever và Client sau đó copy toàn bộ nội dung từ file SQLServerConn.java mới trong thư mục DAO đè lên code cũ.

Tương tự với Client.java

Hướng dẫn sử dụng myProtocol (dùng ở vai trò là client).

Giao thức myProtocol dùng để gởi request lên server và nhận kết quả trả về từ server.

myProtocol có định dạng như sau:



Giao thức myProtocol bao gồm các request sau đây:

Login: <LOGIN><user\_name password IP\_addr>

Register: <REGISTER><user\_name password>

Logout: <LOGOUT><user\_name>

Chấp nhận Kết bạn: <ACCEPT><user\_name friend\_name>

Tìm bạn: <SEARCH><user\_name>

Lấy danh sách bạn: <SEARCH><user\_name>

Gởi lời mời kết bạn: <REQUEST><user\_name friend\_name>

Xóa lời mời kết bạn: <DELETE><user\_name friend\_name>

Cách sử dụng:

Tạo kết nối giữa client vs server.

conn = new Socket(InetAddress.getLocalHost(), 9000);

Tạo đối tượng writer để ghi request và gởi đến server.

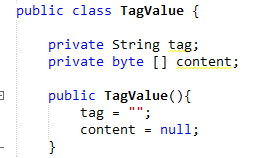
writer = new TagWriter(conn.getOutputStream());

Tạo đối tượng reader để đọc kết quả trả về từ server.

reader = new TagReader(conn.getInputStream());



Tạo đối tượng TagValue (định dạng request để gởi đến server)



Class TagValue có 2 field là String tag (loại Request LOGIN/ LOGOUT/ ...) và byte[] content (nội dung gởi đi)

Ví dụ: client muốn xác thực login đến server thì gởi theo định dạng:

<LOGIN><user\_name password IP\_addr>

String tag = “<LOGIN>”;

byte[] content = “huy 1 10.1.2.3”.getBytes();

/\* huy là user\_name

\* 1 là password

\* 10.1.2.3 là địa chỉ IP của máy trong mạng local

\*/

Cuối cùng tạo đối tượng TagValue để gởi đến server

TagValue tv = new TagValue(tag, content);

Gởi đến server bằng cách:

writer.writeTag(tv);

writer.flush(); // đẩy TagValue đi đến server.

Để nhận kết quả của server trả về:

TagValue tv\_recive = reader.getTagValue();

Dùng tv\_recive.getTag() : String để lấy thông tin của Tag để biết thành công (tag là <SUCCESS>) hay thất bại (tag là <FAIL>)

Đối với các request như LOGIN, REGISTER, ACCEPT, REQUEST, DELETE thì server sẽ trả về <SUCCESS><> hoặc <FAIL><> mà không có content.

Với tag SEARCH (tìm kiếm User theo tên của user đó) thì server trả về <FAIL><> khi không có user đó hoặc <SUCCESS><userID userName IP\_addr Status>

Đối với kết quả <SUCCESS><userID userName IP\_addr Status> trả về từ server.

Dùng byte[] content = tv.getContent()

Dùng hàm getUsers(byte[] content) : List<User> để lấy thông tin user đã tìm thành công.

List<User> usrs = getUserss(content)

Vì chỉ tìm có 1 người nên danh sách usrs chỉ có 1 phần tử

Với Tag FIND\_FRIEND với FIND\_REQUEST thì khi dùng List<User> usrs = getUserss(content) sẽ trả về 1 danh sách có thể số phần từ >= 1.

* Để tạo một group chat/ thêm 1 user vào group:

<CREATE><group\_name user\_name>

Server trả về <SUCCESS><> hoặc <FAIL><>

* Để thoát khỏi một group chat

<LEAVE><group\_name user\_name>

Server trả về <SUCCESS><> hoặc <FAIL><>

* Để lấy tên các group chat mà mình tham gia

<GET\_MY\_GROUP><user\_name>

Server trả về <SUCCESS><name\_group1|name\_group2|name\_group3>

Dùng hàm private String[] getGroupName(byte [] content) với tham

Số content truyền vào là TagValue.getContent() để lấy danh sách

Tên group

Hoặc Server trả về <FAIL><> nếu không có tham gia vào group nào.

* Để lấy thông tin các thành viên trong group dùng

<GET\_MEMBER><group\_name>

Server trả về <SUCCESS> nếu tồn tại thành viên trong group và dùng hàm List<User> getUsers(byte [] content) để lấy danh sách các user.

Server trả về <FAIL> nếu k có thành viên nào