

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package chat.client;
7
8  import chat.protocol.Message;
9  import java.io.BufferedReader;
10
11
12
13  /**
14   * Thread client d'écoute de socket.
15   * @author Nico
16   */
17  public class ClientThread extends Thread{
18      /**
19       * Référence sur le client correspondant.
20       */
21      Client client;
22
23      /**
24       * Flux de lecture de message.
25       */
26      BufferedReader socIn;
27
28      /**
29       * Constructeur.
30       * @param c Référence sur le client correspondant.
31       * @param br Flux de lecture de message.
32       */
33      public ClientThread(Client c,BufferedReader br) {
34          client = c;
35          socIn = br;
36      }
37
38      /**
39       * Méthode exécutée par le thread. Lit les messages provenant du serveur et
40       * en notifie le "client".
41       * Traite les lignes que le BufferedReader socIn (de la socket) reçoit du
42       * serveur et analyse ce qu'elle reçoit et fait le traitement approprié
43       * selon si c'est un message informatif(quelqu'un s'est connecté ou
44       * déconnecté) ou un message d'un client.
45       */
46      @Override
47      public void run(){
```

```
48         try {
49             String line;
50             while (true) {
51                 line = socIn.readLine();
52                 treatLine(line);
53             }
54         } catch (Exception e) {
55             System.err.println("ClientThread exception : " +
56                                 e.toString());
57         }
58
59         /**
60          * Methode traitant une ligne recue par le flux d'entrée "socIn".
61          * @param line Ligne lue.
62          */
63         private void treatLine(String line){
64             if(line.contains("SIGNIN")) {
65                 String[] splits = line.split("SIGNIN ");
66                 client.newUserConnected(splits[1]);
67             } else if(line.contains("SIGNOUT")) {
68                 String[] splits = line.split("SIGNOUT ");
69                 client.userDisconnected(splits[1]);
70             } else if(line.contains("MESSAGE FROM") &&
71                         line.contains("TO") && line.
72                         contains("CONTENT")) {
73                 String[] splits1 = line.split("MESSAGE FROM ");
74                 String[] splits2 = splits1[1].split(" TO ");
75                 String[] splits3 = splits2[1].split(" CONTENT ");
76                 String sender = splits2[0];
77                 String receiver = splits3[0];
78                 String textMessage = splits3[1];
79                 client.displayMessage(new
80                                     Message(sender,receiver,textMessage));
81             }
82         }
```