

C++ Code

```
1  //
2  // Client C++ pour communiquer avec un serveur TCP
3  // eric lecolinet - telecom paristech - 2016/2020
4  //
5
6  #include <iostream>
7  #include <string>
8  #include <algorithm>
9  #include "ccsocket.h"
10
11 static const std::string HOST = "127.0.0.1";
12 static const int PORT = 3331;
13
14 /**
15  /// Lit une requeste depuis le Terminal, envoie cette requeste au serveur,
16  /// recupere sa reponse et l'affiche sur le Terminal.
17  /// Noter que le programme bloque si le serveur ne repond pas.
18  /**
19
20 int main() {
21     Socket sock;
22     SocketBuffer sockbuf(sock);
23
24     int status = sock.connect(HOST, PORT);
25
26     if (status < 0) {
27         switch (status) {
28             case Socket::Failed:
29                 std::cerr << "Client: Couldn't reach host " << HOST << ":" << PORT <<
29 std::endl;
30                 return 1;
31             case Socket::UnknownHost:
32                 std::cerr << "Client: Couldn't find host " << HOST << ":" << PORT <<
32 std::endl;
33                 return 1;
34             default:
35                 std::cerr << "Client: Couldn't connect host " << HOST << ":" << PORT <<
35 std::endl;
36                 return 1;
37         }
38     }
39
40     std::cout << "Client connected to " << HOST << ":" << PORT << std::endl;
41
42     while (std::cin) {
43         std::cout << "Request: ";
44         std::string request, response;
45
46         std::getline(std::cin, request);
47         if (request == "quit") return 0;
```

```
48     // Envoyer la requete au serveur
49     if (sockbuf.writeLine(request) < 0) {
50         std::cerr << "Client: Couldn't send message" << std::endl;
51         return 2;
52     }
53
54
55     // Recuperer le resultat envoye par le serveur
56     if (sockbuf.readLine(response) < 0) {
57         std::cerr << "Client: Couldn't receive message" << std::endl;
58         return 2;
59     }
60
61     // Le serveur remplace les '\n' par des ';' car '\n' sert a indiquer la
62     // fin d'un message entre le client et le serveur
63     // On fait ici la transformation inverse
64     std::replace(response.begin(), response.end(), ';', '\n');
65
66     std::cout << "Response: " << response << std::endl;
67 }
68 }
69 }
```