

Code client – Client.java

```
1 import java.io.DataInputStream;
2 import java.net.Socket;
3
4 public class Client
5 {
6     private static Socket socket;
7
8     /*
9      * Application client
10     */
11     public static void main(String[] args) throws Exception
12     {
13         // Adresse et port du serveur
14         String serverAddress = "127.0.0.1";
15         int port = 5000;
16
17         // Création d'une nouvelle connexion avec le serveur
18         socket = new Socket(serverAddress, port);
19
20         System.out.format("The server is running on %s:%d\n", serverAddress, port);
21
22         // Création d'un canal entrant pour recevoir les messages envoyés par le serveur
23         DataInputStream in = new DataInputStream(socket.getInputStream());
24
25         // Attente de la réception d'un message envoyé par le serveur sur le canal
26         String helloMessageFromServer = in.readUTF();
27         System.out.println(helloMessageFromServer);
28
29         // Fermeture de la connexion avec le serveur
30         socket.close();
31     }
32 }
```

Code serveur – Server.java

```
1 import java.io.DataOutputStream;
2 import java.io.IOException;
3 import java.net.InetAddress;
4 import java.net.InetSocketAddress;
5 import java.net.ServerSocket;
6 import java.net.Socket;
7
8 public class Server
9 {
10     private static ServerSocket listener;
11
12     /*
13      * Application Serveur
14      */
15     public static void main(String[] args) throws Exception
16     {
17         // Compteur incrémenté à chaque connexion d'un client au serveur
18         int clientNumber = 0;
19
20         // Adresse et port du serveur
21         String serverAddress = "127.0.0.1";
22         int serverPort = 5000;
23
24         // Création de la connexion pour communiquer avec les clients
25         listener = new ServerSocket();
26         listener.setReuseAddress(true);
27         InetAddress serverIP = InetAddress.getByName(serverAddress);
28
29         // Association de l'adresse et du port à la connexion
30         listener.bind(new InetSocketAddress(serverIP, serverPort));
31
32         System.out.format("The server is running on %s:%d\n", serverAddress, serverPort);
33
34         try
35         {
36             /*
37              * À chaque fois qu'un nouveau client se connecte, on exécute la fonction
38              * Run() de l'objet ClientHandler.
39              */
40             while (true)
41             {
42                 // Important : la fonction accept() est bloquante : attend qu'un prochain client se connecte
43                 // Une nouvelle connexion : on incrémente le compteur clientNumber
44                 new ClientHandler(listener.accept(), clientNumber++).start();
45             }
46         }
47         finally
48         {
49             // Fermeture de la connexion
50             listener.close();
51         }
52     }
53 }
```

```
54
55 /*
56  * Une thread qui se charge de traiter la demande de chaque client
57  * sur un socket particulier.
58  */
59 private static class ClientHandler extends Thread
60 {
61     private Socket socket;
62     private int clientNumber;
63
64     public ClientHandler(Socket socket, int clientNumber)
65     {
66         this.socket = socket;
67         this.clientNumber = clientNumber;
68         System.out.println("New connection with client#" + clientNumber + " at " + socket);
69     }
70
71     /*
72      * Une thread se charge d'envoyer au client un message de bienvenue
73      */
74     public void run()
75     {
76         try
77         {
78             // Création d'un canal sortant pour envoyer des messages au client
79             DataOutputStream out = new DataOutputStream(socket.getOutputStream());
80
81             // Envoie d'un message au client
82             out.writeUTF("Hello from server - you are client#" + clientNumber);
83
84         } catch (IOException e)
85         {
86             System.out.println("Error handling client# " + clientNumber + ": " + e);
87         }
88         finally
89         {
90             try
91             {
92                 // Fermeture de la connexion avec le client
93                 socket.close();
94             } catch (IOException e)
95             {
96                 System.out.println("Couldn't close a socket, what's going on?");
97             }
98             System.out.println("Connection with client# " + clientNumber + " closed");
99         }
100     }
101 }
102 }
```

Sorties sur les consoles

Client

Serveur

```
The server is running on 127.0.0.1:5000
Hello from server - you are client#0
```

```
The server is running on 127.0.0.1:5000
New connection with client#0 at Socket[addr=/127.0.0.1,port=59606,localport=5000]
Connection with client# 0 closed
```