Architecture Client-Serveur

Modèle client/serveur

Sockets (Code client : Langage Java)

```
import java.io.DataInputStream;
import java.net.Socket;
// Application client
public class Client {
        private static Socket socket;
        public static void main(String[] args) throws Exception {
                // Adresse et port du serveur
                String serverAddress = "127.0.0.1";
                int port = 5000;
                // Création d'une nouvelle connexion aves le serveur
                socket = new Socket(serverAddress, port);
                System.out.format("Serveur lancé sur [%s:%d]", serverAddress, port);
                // Céatien d'un canal entrant pour recevoir les messages envoyés, par le serveur
                DataInputStream in = new DataInputStream(socket.getInputStream());
                // Attente de la réception d'un message envoyé par le, server sur le canal
                String helloMessageFromServer = in.readUTF();
                System.out.println(helloMessageFromServer);
                // fermeture de La connexion avec le serveur
                socket.close();
        }
}
```

Architecture Client-Serveur

Modèle client/serveur

Code serveur : Langage Java

```
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.ServerSocket;
public class Serveur {
        private static ServerSocket Listener;
        // Application Serveur
        public static void main(String[] args) throws Exception {
                // Compteur incrémenté à chaque connexion d'un client au serveur
                int clientNumber = 0;
                // Adresse et port du serveur
                // String serverAddress = "127.0.0.1";
                // int serverPort = 5000;
            // Création de la connexien pour communiquer ave les, clients
            Listener = new ServerSocket();
            Listener.setReuseAddress(true);
            InetAddress serverIP = InetAddress.getByName(serverAddress);
            // Association de l'adresse et du port à la connexien
            Listener.bind(new InetSocketAddress(serverIP, serverPort));
            System.out.format("The server is running on %s:%d%n", serverAddress, serverPort);
          try {
              // À chaque fois qu'un nouveau client se, connecte, on exécute la fonstion
              // run() de l'objet ClientHandler
              while (true) {
                        // Important : la fonction accept() est bloquante: attend qu'un prochain client se
connecte
                        // Une nouvetle connection : on incémente le compteur clientNumber
                        new ClientHandler(Listener.accept(), clientNumber++).start();
                }
           } finally {
                        // Fermeture de la connexion
                        Listener.close();
            } } }
```

```
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
public class ClientHandler extends Thread { // pour traiter la demande de chaque client sur un socket particulier
                private Socket socket;
                private int clientNumber;
                public ClientHandler(Socket socket, int clientNumber) {
                                this.socket = socket;
                                this.clientNumber = clientNumber;
                                System.out.println("New connection with client#" + clientNumber + " at" + socket);
                }
                public void run() {  // Création de thread qui envoi un message à un client
                                try {
                    DataOutputStream out = new DataOutputStream(socket.getOutputStream()); // création de canal d'envoi
                                    out.writeUTF("Hello from server - you are client#" + clientNumber); // envoi de message
                                } catch (IOException e) {
                                    System.out.println("Error handling client# " + clientNumber + ": " + e);
                                } finally {
                                                try {
                                                   socket.close();
                                                } catch (IOException e) {
                                                    System.out.println("Couldn't close a socket, what's going on?");
                                                }
                                                 System.out.println("Connection with client# " + clientNumber + " closed");
                                }
                }
```

Les sockets

Sorties sur les consoles

- Client

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

- Serveur

```
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
```