

## Application Client Serveur Echo

### 1. Programme du serveur Echo

```
import java.io.*;
import java.net.*;

....

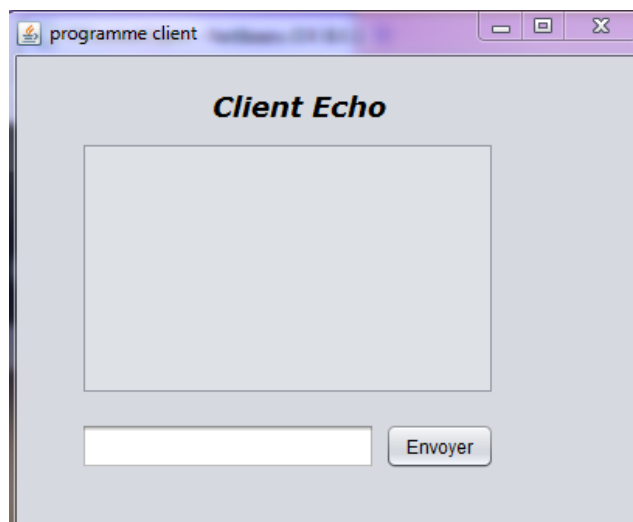
class job implements Runnable{
    Socket s;
    job (Socket s){ this.s=s;}
    public void run(){
        DataInputStream Sin= new DataInputStream(s.getInputStream());
        DataOutputStream Sout= new DataOutputStream(s.getOutputStream());
        String msg = Sin.readUTF();
        while (!msg.equalsIgnoreCase("Fin")){
            System.out.println ("Le Client a envoyé " + msg);
            Sout.writeUTF(msg.toUpperCase());
            msg = Sin.readUTF();
        }
        System.out.println ("Session de communication terminée");
        s.close();}}

...

public static void main(String[] args) {
    ServerSocket svr = new ServerSocket(5007);
    System.out.println ("Le serveur est disponible") ;
    while (true){
        Socket service = svr.accept();
        Runnable Tache= new job(service);
        Thread t = new Thread(Tache);
        t.start();
        System.out.println ("Le serveur a crée " + t.getName());}}}
```

### 2. Interface coté client

Créer une interface avec un label, TextArea (avec saisie désactivée), button.



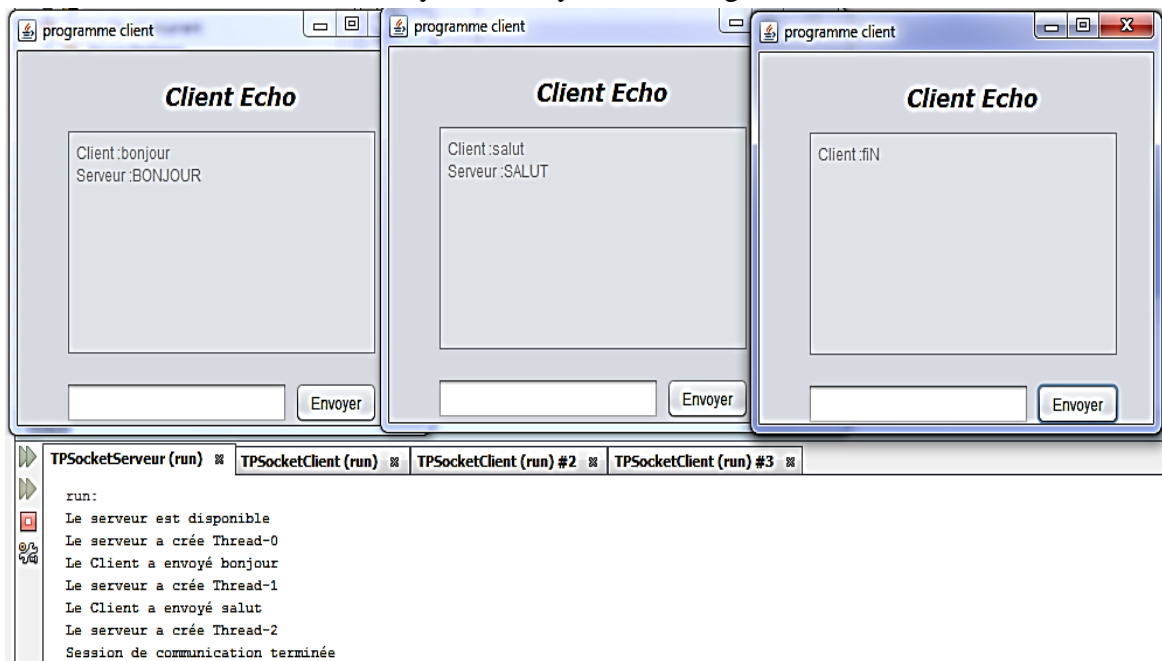
## Application Client Serveur Echo

### 3. Code du client

```
public class Client extends javax.swing.JFrame {
    static Socket monClient ;
    static DataOutputStream Cout;
    static DataInputStream Cin;
    ....
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            String ligne;
            ligne = jTextField1.getText();
            jTextField1.setText("");
            jTextArea1.append("Client :"+ligne+ "\n");
            Cout.writeUTF(ligne);
            if (ligne.equalsIgnoreCase("Fin")){ monClient.close();}
            else{
                jTextArea1.append("Serveur :"+ Cin.readUTF()+ "\n");}
        } catch (IOException ex) {
            jTextArea1.append("connexion perdue !!"); }}
        ....
    public static void main(String args[]) {
        monClient = new Socket("localhost",5007);
        Cin= new DataInputStream(monClient.getInputStream());
        Cout= new DataOutputStream(monClient.getOutputStream());}
}
```

### 4. Exécution

- ✓ Essayer de lancer plusieurs clients
- ✓ Envoyer des messages
- ✓ Envoyer « fin » ....
- ✓ Fermer le serveur et essayer d'envoyer un message ...



**Question Supplémentaire** modifier le serveur pour qu'il envoie au client après « fin » le nombre des messages qu'il a reçue de la part de ce client.