

## Fichier secure serveur tcp pour ssl en JAVA

```
import java.io.*; import java.net.*; import javax.net.ssl.SSLServerSocket;
import javax.net.ssl.SSLServerSocketFactory; import javax.net.ssl.SSLSocket;

public class secureserveurtcp {
    private static int port;
    public static void main (String [] args) throws Exception {
        boolean boucle = true;
        Reader readersoc;
        PrintStream a_envoye=null;
        Socket soc;
        String line;
        if(args.length != 1) {
            System.out.println("usage : java secureserveurtcp port"); System.exit(0); }
        port = Integer.parseInt(args[0]);
        try {
            SSLServerSocketFactory sslserversocketfactory =(SSLServerSocketFactory)
SSLServerSocketFactory.getDefault();
            SSLServerSocket sslserversocket =(SSLServerSocket)
sslserversocketfactory.createServerSocket(port);
            System.out.println("La socket serveur securise est cree");

while (true) {
    boucle = true;
    SSLSocket sslsocket = (SSLSocket) sslserversocket.accept();
    readersoc = new InputStreamReader(sslsocket.getInputStream());
    a_envoye = new PrintStream(sslsocket.getOutputStream());
    BufferedReader datarecu = new BufferedReader (readersoc);

    while (boucle) {
        line = datarecu.readLine();
        System.out.println("Vous avez saisi : " + line);
        if(line.equals("FIN")) {
            boucle = false;
            line = null;
            sslsocket.close(); }
        else {
            StringBuffer lineReversed = (new StringBuffer(line)).reverse();
            a_envoye.println(lineReversed); }
        }
    } catch (Exception exception) {
        exception.printStackTrace();} }}}
```

## Fichier secure client tcp pour ssl en JAVA

```
import java.io.*;
import java.net.*;
import javax.net.ssl.SSLSocket;
import javax.net.ssl.SSLSocketFactory;

public class secureclienttcp
{
    private static int port;
    public static void main (String [] args) throws Exception
    {
        String adresse, line, lineReversed;
        Reader readSoc;
        PrintStream a_envoye=null;
        if(args.length != 2) {
            System.out.println("usage : java secureclienttcp nom_serveur port");
            System.exit(0); }
        adresse = args[0];
        port = Integer.parseInt(args[1]);

        try {
            SSLSocketFactory sslsocketfactory =
(SSLSocketFactory)SSLSocketFactory.getDefault();
            SSLSocket sslsocket =
(SSLSocket)sslsocketfactory.createSocket(adresse, port);
            Reader reader = new InputStreamReader(System.in);
            BufferedReader keyboard = new BufferedReader(reader);
            a_envoye = new PrintStream(sslsocket.getOutputStream());
            readSoc = new InputStreamReader(sslsocket.getInputStream());
            BufferedReader RecuSoc = new BufferedReader (readSoc);
            while (true) {
                System.out.println("Entrez une ligne de texte : ");
                line = keyboard.readLine();
                a_envoye.println(line);
                // si on a tape "FIN" on quitte le client
                if(line.equals("FIN")) break;
                lineReversed = RecuSoc.readLine();
                System.out.println("Recu : " + lineReversed); }
            // fermeture de la socket
            sslsocket.close();
        } catch (Exception exception) {exception.printStackTrace();} }}}
```

## Fichier client tcp socket pour serveur https en JAVA

```
import javax.net.ssl.SSLSocket;
import javax.net.ssl.SSLSocketFactory;
import java.io.*;

public class Httpsclient {
    public static void main(String[] arstring) {

        try {
            SSLSocketFactory sslsocketfactory =
(SSLSocketFactory)SSLSocketFactory.getDefault();
            SSLSocket sslsocket =
(SSLSocket)sslsocketfactory.createSocket("nom_serveur_web", 443);
            System.out.println("connexion passee");

            PrintStream a_envoye = new PrintStream(sslsocket.getOutputStream());
            InputStreamReader readSoc = new
InputStreamReader(sslsocket.getInputStream());
            BufferedReader RecuSoc = new BufferedReader (readSoc);

            String line="GET / HTTP/1.1\r\nHost:nom\r\n\r\n";
            A_envoye.println(line);
            String string = null;
            while ((string = RecuSoc.readLine()) != null) {
                System.out.println(string);
            }
        } catch (Exception exception) {
            exception.printStackTrace();
        }
    }
}
```

## Fichier client utilisant httpsURLConnection en JAVA

```
import java.net.MalformedURLException;
import java.net.URL;import java.security.cert.Certificate;
import java.io.*;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLPeerUnverifiedException;

public class Httpsclient{

    public static void main(String[] args)
    {
        new Httpsclient().testIt();
    }

    private void testIt(){

        String https_url = "https://a_remplir/";
        URL url;
        try {
            url = new URL(https_url);
            HttpURLConnection con = (HttpURLConnection)url.openConnection();

            //dump all the content
            print_content(con);
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    private void print_content(HttpURLConnection con){
        if(con!=null){
            try {
                System.out.println("***** Content of the URL *****");
                BufferedReader br = new BufferedReader(
                    new InputStreamReader(con.getInputStream()));

                String input;
                while ((input = br.readLine()) != null){
                    System.out.println(input);
                }
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```