# JAVA網路

## 網路常識

TCP(Transmission Control Protocol)：Socket程式，雙方建立可信連接，才進行資料傳輸。

UDP(User Datagram Protocol)：連接不可信。

伺服器接收Socket：ServerSocket

用戶端連接Socket：Socket

**import** java.io.PrintStream;

**import** java.net.ServerSocket;

**import** java.net.Socket;

**public** **class** ServerDemo {

**public** **static** **void** main(String[] args) **throws** Exception {

ServerSocket server = **new** ServerSocket(1234);

System.***out***.println("伺服器啟動");

Socket client = server.accept();

String str = "Hello";

PrintStream out = **new** PrintStream(client.getOutputStream());

out.print(str);

out.close();

client.close();

server.close();

}

}

開啟Telnet，輸入open localhost 1234，即可看到server傳來的信息。

**import** java.io.PrintStream;

**import** java.net.ServerSocket;

**import** java.net.Socket;

**import** java.util.Scanner;

**public** **class** ClientDemo {

**public** **static** **void** main(String[] args) **throws** Exception {

Socket client = **new** Socket("localhost", 1234);

Scanner scan = **new** Scanner(client.getInputStream());

scan.useDelimiter("\n");

**if** (scan.hasNext()) {

System.***out***.println(scan.next());

}

client.close();

}

}

## ECHO

客戶端向伺服器發送信息，伺服器處理後，在信息前加”ECHO”，

import java.io.IOException;

import java.io.PrintStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Scanner;

public class EchoSingle {

public static void main(String[] args) throws IOException {

ServerSocket server = new ServerSocket(1234);

Socket client = server.accept();

boolean flag = true;

while (flag) {

Scanner clientScan = new Scanner(client.getInputStream());

if (clientScan.hasNext()) {

String msg = clientScan.next();

if ("bye".equalsIgnoreCase(msg)) {

flag = false;

}

PrintStream clientPrint = new PrintStream(client.getOutputStream());

clientPrint.println("ECHO："+msg);

}

}

client.close();

server.close();

}

}

## 多執行緒

使用多執行緒執行。

import java.io.IOException;

import java.io.PrintStream;

import java.net.Socket;

import java.util.Scanner;

public class EchoThread implements Runnable {

private Socket client = null;

public EchoThread(Socket client){

this.client = client;

}

@Override

public void run() {

// TODO Auto-generated method stub

boolean flag = true;

try {

while (flag) {

Scanner clientScan = new Scanner(client.getInputStream());

if (clientScan.hasNext()) {

String msg = clientScan.next();

if ("bye".equalsIgnoreCase(msg)) {

flag = false;

}

PrintStream clientPrint = new PrintStream(client.getOutputStream());

clientPrint.println("ECHO："+msg);

}

}

this.client.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

import java.io.IOException;

import java.io.PrintStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Scanner;

public class EchoServer {

public static void main(String[] args) throws IOException{

ServerSocket server = new ServerSocket(1234);

boolean flag = true;

while (flag) {

Socket client = server.accept();

new Thread(new EchoThread(client)).start();

}

server.close();

}

}