|  |  |
| --- | --- |
| H:\LOGO-NXV\Hai__090908__02_1_den.jpg | Faculty of Information Technology  H A N O I U N I V E R S I T Y |

61FIT3NPR – Network Programming   
Tutorial

Java TCP Socket

1. Exercise 1: TCP socket

import java.io.\*;  
import java.net.\*;  
  
class TCPServer {  
 public static void main(String argv[]) throws Exception {  
 String clientSentence;  
 String capitalizedSentence;  
 ServerSocket welcomeSocket = new ServerSocket(6789);  
 System.*out*.println("Server is waiting to accept user... ");  
  
 //while(true) {  
 Socket connectionSocket = welcomeSocket.accept();  
 System.*out*.println("Accept a client!");  
  
 BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));  
 DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());  
  
 while(true) {  
 clientSentence = inFromClient.readLine();  
 capitalizedSentence = clientSentence.toUpperCase() + "\n";  
  
 outToClient.writeBytes(capitalizedSentence);  
 }  
 }  
}

import java.io.\*;  
import java.net.\*;  
  
public class TCPClient {  
 public static void main(String argv[]) throws Exception {  
 String sentence;  
 String modifiedSentence;  
 BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.*in*));  
  
 //while(true) {  
 Socket clientSocket = new Socket("localhost", 6789);  
 DataOutputStream outToServer = new DataOutputStream (clientSocket.getOutputStream());  
 BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));  
  
  
 while(true) {  
 System.*out*.println("Please enter your message");  
 sentence = inFromUser.readLine();  
 outToServer.writeBytes(sentence + '\n');  
 modifiedSentence = inFromServer.readLine();  
 System.*out*.println("FROM SERVER: " + modifiedSentence);  
 //clientSocket.close();  
 }  
 }  
}

1. Exercise 2: Another client/server program

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.io.OutputStreamWriter;

**import** java.net.ServerSocket;

**import** java.net.Socket;

**public** **class** SimpleServerProgram {

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

ServerSocket listener = **null**;

String line;

BufferedReader is;

BufferedWriter os;

Socket socketOfServer = **null**;

**try** {

listener = **new** ServerSocket(9999);

} **catch** (IOException e) {

System.***out***.println(e);

System.*exit*(1);

}

**try** {

System.***out***.println("Server is waiting to accept user...");

socketOfServer = listener.accept();

System.***out***.println("Accept a client!");

is = **new** BufferedReader(**new** InputStreamReader(socketOfServer.getInputStream()));

os = **new** BufferedWriter(**new** OutputStreamWriter(socketOfServer.getOutputStream()));

**while** (**true**) {

line = is.readLine();

os.write(">> " + line);

os.newLine();

os.flush();

**if** (line.equals("QUIT")) {

os.write(">> OK");

os.newLine();

os.flush();

**break**;

}

}

} **catch** (IOException e) {

System.***out***.println(e);

e.printStackTrace();

}

System.***out***.println("Sever stopped!");

}

}

**import** java.io.\*;

**import** java.net.\*;

**public** **class** SimpleClientDemo {

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

**final** String serverHost = "localhost";

Socket socketOfClient = **null**;

BufferedWriter os = **null**;

BufferedReader is = **null**;

**try** {

socketOfClient = **new** Socket(serverHost, 9999);

os = **new** BufferedWriter(**new** OutputStreamWriter(socketOfClient.getOutputStream()));

is = **new** BufferedReader(**new** InputStreamReader(socketOfClient.getInputStream()));

} **catch** (UnknownHostException e) {

System.***err***.println("Don't know about host " + serverHost);

**return**;

} **catch** (IOException e) {

System.***err***.println("Couldn't get I/O for the connection to " + serverHost);

**return**;

}

**try** {

os.write("HELLO");

os.newLine();

os.flush();

os.write("I am Tom Cat");

os.newLine();

os.flush();

os.write("QUIT");

os.newLine();

os.flush();

String responseLine;

**while** ((responseLine = is.readLine()) != **null**) {

System.***out***.println("Server: " + responseLine);

**if** (responseLine.indexOf("OK") != -1) {

**break**;

}

}

os.close();

is.close();

socketOfClient.close();

} **catch** (UnknownHostException e) {

System.***err***.println("Trying to connect to unknown host: " + e);

} **catch** (IOException e) {

System.***err***.println("IOException: " + e);

}

}

}

1. Exercise 3: Another client/server program

**Write a program that has 2 sides: client and server, they communicate to each other using TCP socket.**

**A server that:**

- can accept connect from client.

- receives integer number n from client.

- calculates the value of square of n.

- and sends back square of n to client.

**A client that:**

- can connect to server.

- sends an integer number n to server.

- and receives square of n from server.