|  |  |
| --- | --- |
| 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

Sending file using Java TCP Sockets

**Example 1: Transfer the File “Client TCP Socket to Server TCP Socket” in Java**

https://www.geeksforgeeks.org/transfer-the-file-client-socket-to-server-socket-in-java/

This document describes a one-way client and Server Setup where a client connects, and sends the file to the server and the server writes the file in another location with a different name. It means we send the file using the server socket.

## **Socket Programming in Java**

Socket Programming is used for communicating different JRE in the different networks in java. In simple words, we can say socket programming is connecting two nodes in different networks and communicating two each other.

### **File Transfer Implementation in Java Socket**

In this example, we will create client.java class in this class we make the Socket object and define the server socket port number for communication. In this class, we select which file send over the network.

### **Client**

* To connect to another machine we need two pieces of information first one is the IP address and the second one is the port number.
* In our case, we are using localhost and the port is 900
* We make a Socket object using the java.net package

#### **Example**

Socket ob = new Socket(ip,port \_ number)

Now we call the “sendFile” method with the parameter of a file path and we open the file and send the file to the server socket using DataOutputStream Class

|  |
| --- |
| **import** java.io.\*;  **import** java.net.Socket;    **public** **class** Client {  **private** **static** DataOutputStream dataOutputStream = **null**;  **private** **static** DataInputStream dataInputStream = **null**;    **public** **static** **void** main(String[] args)      {          // Create Client Socket connect to port 900  **try** (Socket socket = **new** Socket("localhost", 900)) {              dataInputStream = **new** DataInputStream(                  socket.getInputStream());              dataOutputStream = **new** DataOutputStream(                  socket.getOutputStream());              System.out.println(                  "Sending the File to the Server");            // Call SendFile Method            sendFile(                  "/home/dachman/Desktop/Program/gfg/JAVA\_Program/File Transfer/txt.pdf");                dataInputStream.close();              dataInputStream.close();          }  **catch** (Exception e) {              e.printStackTrace();          }      }        // sendFile function define here  **private** **static** **void** sendFile(String path)  **throws** Exception      {  **int** bytes = 0;          // Open the File where he located in your pc          File file = **new** File(path);          FileInputStream fileInputStream              = **new** FileInputStream(file);            // Here we send the File to Server          dataOutputStream.writeLong(file.length());          // Here we  break file into chunks  **byte**[] buffer = **new** **byte**[4 \* 1024];  **while** ((bytes = fileInputStream.read(buffer))                 != -1) {            // Send the file to Server Socket            dataOutputStream.write(buffer, 0, bytes);              dataOutputStream.flush();          }          // close the file here          fileInputStream.close();      }  } |

### **Server**

Here,  we define the ServerSocket object using the ServerSocket class.

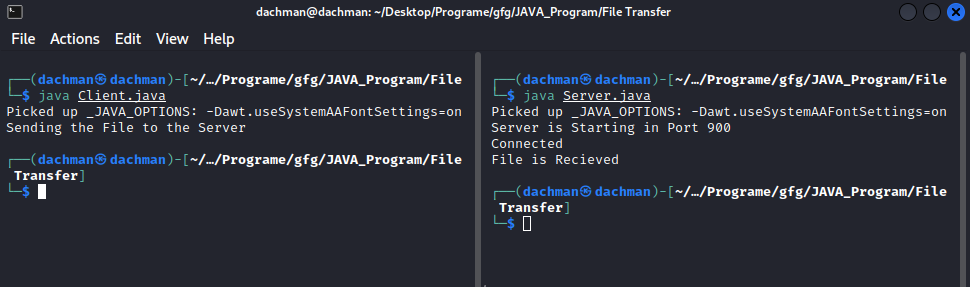
**Example:**

ServerSocket server = new ServerSocket(port\_number)

* when the client sent the request to the server socket. we will call the “receiveFile” method
* We receive the file from the client socket and read the file using the data input stream class
* In this method, we will change the file name and the location of the file. write the file using FileOutputStream Class.
* Java

|  |
| --- |
| **import** java.io.DataInputStream;  **import** java.io.DataOutputStream;  **import** java.io.FileOutputStream;  **import** java.net.ServerSocket;  **import** java.net.Socket;    **public** **class** Server {    **private** **static** DataOutputStream dataOutputStream = **null**;  **private** **static** DataInputStream dataInputStream = **null**;    **public** **static** **void** main(String[] args)      {          // Here we define Server Socket running on port 900  **try** (ServerSocket serverSocket               = **new** ServerSocket(900)) {              System.out.println(                  "Server is Starting in Port 900");              // Accept the Client request using accept method              Socket clientSocket = serverSocket.accept();              System.out.println("Connected");              dataInputStream = **new** DataInputStream(                  clientSocket.getInputStream());              dataOutputStream = **new** DataOutputStream(                  clientSocket.getOutputStream());              // Here we call receiveFile define new for that              // file              receiveFile("NewFile1.pdf");                dataInputStream.close();              dataOutputStream.close();              clientSocket.close();          }  **catch** (Exception e) {              e.printStackTrace();          }      }        // receive file function is start here    **private** **static** **void** receiveFile(String fileName)  **throws** Exception      {  **int** bytes = 0;          FileOutputStream fileOutputStream              = **new** FileOutputStream(fileName);    **long** size              = dataInputStream.readLong(); // read file size  **byte**[] buffer = **new** **byte**[4 \* 1024];  **while** (size > 0                 && (bytes = dataInputStream.read(                         buffer, 0,                         (**int**)Math.min(buffer.length, size)))                        != -1) {              // Here we write the file using write method              fileOutputStream.write(buffer, 0, bytes);              size -= bytes; // read upto file size          }          // Here we received file          System.out.println("File is Received");          fileOutputStream.close();      }  } |

### Output:



#### **Example 2: Transfer a file from Server TCP Socket to Client TCP Socket**

https://www.rgagnon.com/javadetails/java-0542.html

A client module connects to a server then a file is sent to the client.

This example is very simple with no authentication and hard-coded filename!

First the server module.

import java.io.BufferedInputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.OutputStream;

import java.net.ServerSocket;

import java.net.Socket;

public class SimpleFileServer {

public final static int SOCKET\_PORT = 13267;

// you may change this

public final static String FILE\_TO\_SEND = "c:/temp/source.pdf";

// you may change this

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

FileInputStream fis = null;

BufferedInputStream bis = null;

OutputStream os = null;

ServerSocket servsock = null;

Socket sock = null;

try {

servsock = new ServerSocket(SOCKET\_PORT);

while (true) {

System.out.println("Waiting...");

try {

sock = servsock.accept();

System.out.println("Accepted connection : " + sock);

// send file

File myFile = new File (FILE\_TO\_SEND);

byte [] mybytearray = new byte [(int)myFile.length()];

fis = new FileInputStream(myFile);

bis = new BufferedInputStream(fis);

bis.read(mybytearray,0,mybytearray.length);

os = sock.getOutputStream();

System.out.println("Sending " + FILE\_TO\_SEND + "(" + mybytearray.length + " bytes)");

os.write(mybytearray,0,mybytearray.length);

os.flush();

System.out.println("Done.");

}

finally {

if (bis != null) bis.close();

if (os != null) os.close();

if (sock!=null) sock.close();

}

}

}

finally {

if (servsock != null) servsock.close();

}

}

}

The client module

import java.io.BufferedOutputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.net.Socket;

public class SimpleFileClient {

public final static int SOCKET\_PORT = 13267; // you may change this

public final static String SERVER = "127.0.0.1"; // localhost

public final static String

FILE\_TO\_RECEIVED = "c:/temp/source\_copy.pdf";

// you may change this, I give a

// different name because i don't want to

// overwrite the one used by server...

public final static int FILE\_SIZE = 6022386;

// file size temporary hard coded

// should bigger than the file to be downloaded

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

int bytesRead;

int current = 0;

FileOutputStream fos = null;

BufferedOutputStream bos = null;

Socket sock = null;

try {

sock = new Socket(SERVER, SOCKET\_PORT);

System.out.println("Connecting...");

// receive file

byte [] mybytearray = new byte [FILE\_SIZE];

InputStream is = sock.getInputStream();

fos = new FileOutputStream(FILE\_TO\_RECEIVED);

bos = new BufferedOutputStream(fos);

bytesRead = is.read(mybytearray,0,mybytearray.length);

current = bytesRead;

do {

bytesRead =

is.read(mybytearray, current, (mybytearray.length-current));

if(bytesRead >= 0) current += bytesRead;

} while(bytesRead > -1);

bos.write(mybytearray, 0 , current);

bos.flush();

System.out.println("File " + FILE\_TO\_RECEIVED

+ " downloaded (" + current + " bytes read)");

}

finally {

if (fos != null) fos.close();

if (bos != null) bos.close();

if (sock != null) sock.close();

}

}

}

To try it, first you start the server. You make sure that the file to be sent (as specified in SimpleFileServer) exists! Then you execute the client module.

Part