Java Programming Language SE - 6

Module 16: Networking

Team Emertxe

ORACLE®
Certified Professional
Java SE 6 Programmer

EMERTXE

Objectives

- Develop code to set up the network connection
- Understand the TCP/IP Protocol
- Use ServerSocket and Socket classes for implementation of TCP/IP clients and servers

EMERTXE

Relevance

- How can a communication link between a client machine and a server on the network be established?

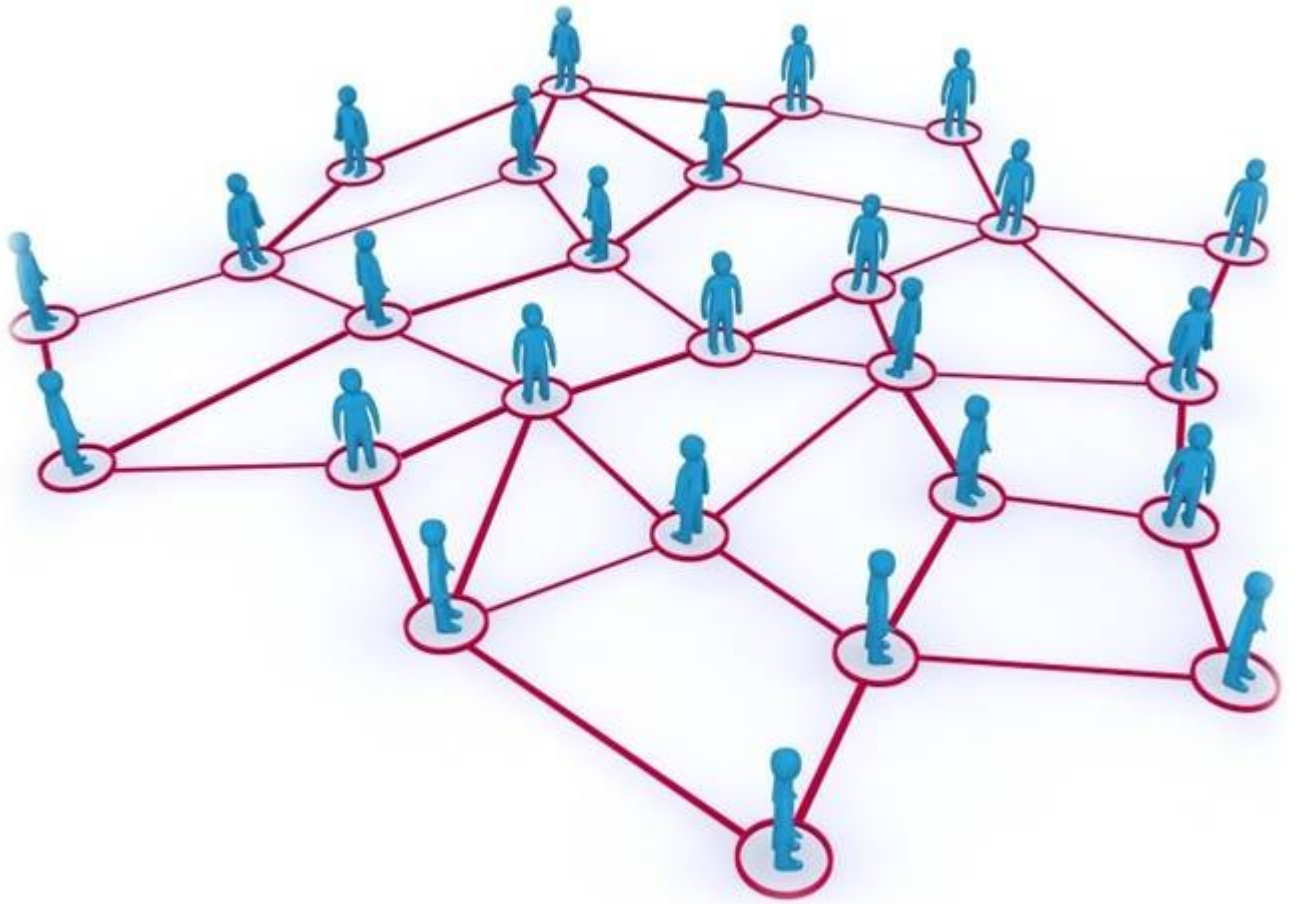
Networking

This section describes networking concepts.

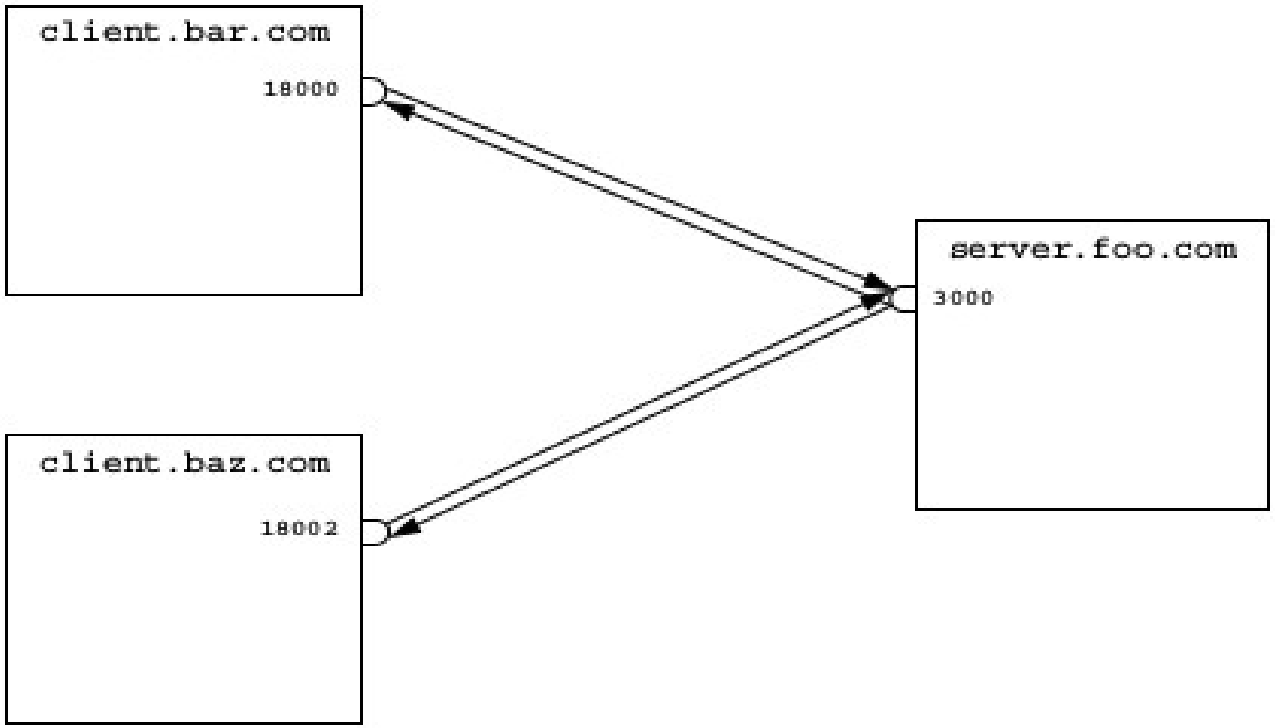
- Sockets
  - Sockets hold two streams: an input stream and an output stream.
  - Each end of the socket has a pair of streams.
- Setting Up the Connection

  Set up of a network connection is similar to a telephone system: One end must dial the other end, which must be listening.

Networking



Networking

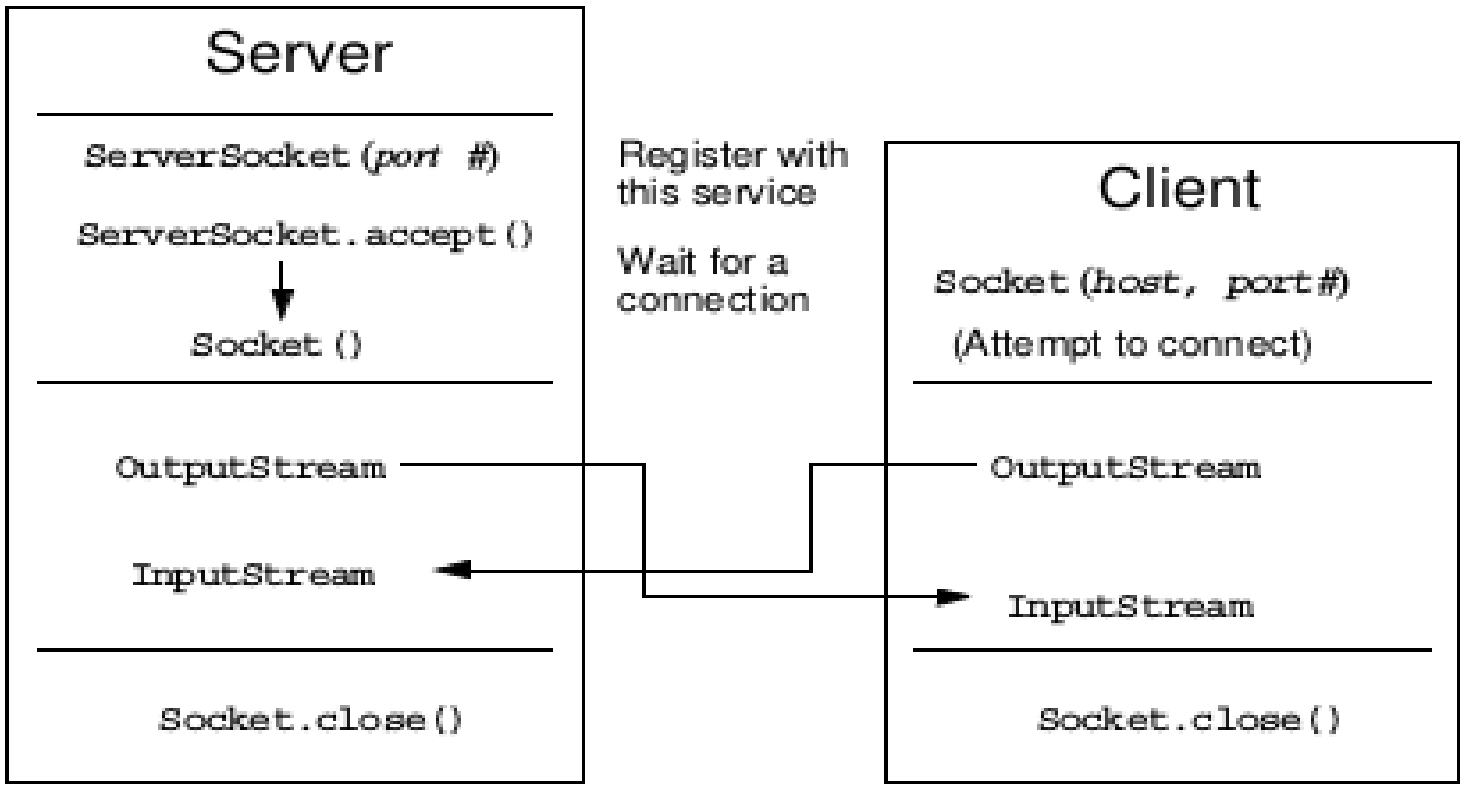


EMERTXE

Networking With Java Technology

- To address the connection, include the following:
  - The address or name of remote machine
  - A port number to identify the purpose at the server
- Port numbers range from 0–65535.

Java Networking Model



Minimal TCP/IP Server

```
import java.net.*;
import java.io.*;
public class SimpleServer {
public static void main(String args[]) {
ServerSocket s = null;
// Register your service on port 5432
try {
s = new ServerSocket(5432);
} catch (IOException e) {
e.printStackTrace();
}
```

Minimal TCP/IP Server

```
// Run the listen/accept loop forever
while (true) {
try {
// Wait here and listen for a connection
Socket s1 = s.accept();
// Get output stream associated with the socket
OutputStream s1out = s1.getOutputStream();
BufferedWriter bw = new BufferedWriter(
new OutputStreamWriter(s1out));
// Send your string!
bw.write("Hello Net World!\n");
```

Minimal TCP/IP Server

```
// Close the connection, but not the server socket
bw.close();
s1.close();
} catch (IOException e) {
e.printStackTrace();
} // END of try-catch
} // END of while(true)
} // END of main method
} // END of SimpleServer program
```

Minimal TCP/IP Client

```
import java.net.*;
import java.io.*;
public class SimpleClient {
public static void main(String args[]) {
try {
// Open your connection to a server, at port 5432
// localhost used here
Socket s1 = new Socket("127.0.0.1", 5432);
// Get an input stream from the socket
InputStream is = s1.getInputStream();
// Decorate it with a "data" input stream
DataInputStream dis = new DataInputStream(is);
```

Minimal TCP/IP Client

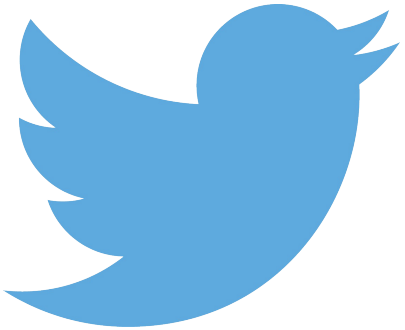
```
// Read the input and print it to the screen
System.out.println(dis.readUTF());
// When done, just close the steam and connection
dis.close();
s1.close();
} catch (ConnectException connExc) {
System.err.println("Could not connect.");
} catch (IOException e) {
// ignore
} // END of try-catch
} // END of main method
} // END of SimpleClient program
```

Stay connected

About us: Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,
No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046
T: +91 80 6562 9666
E: training@emertxe.com

https://www.facebook.com/Emertxe



https://twitter.com/EmertxeTweet

https://www.slideshare.net/EmertxeSlides

EMERTXE

Thank You