Sockets

https://docs.oracle.com/javase/tutorial/networking/sockets/index.html

Sockets

Sockets are one end of a two-way connection between programs running on a network.

Programs use sockets to communicate (send data back and forth) over the Internet (or local networks).

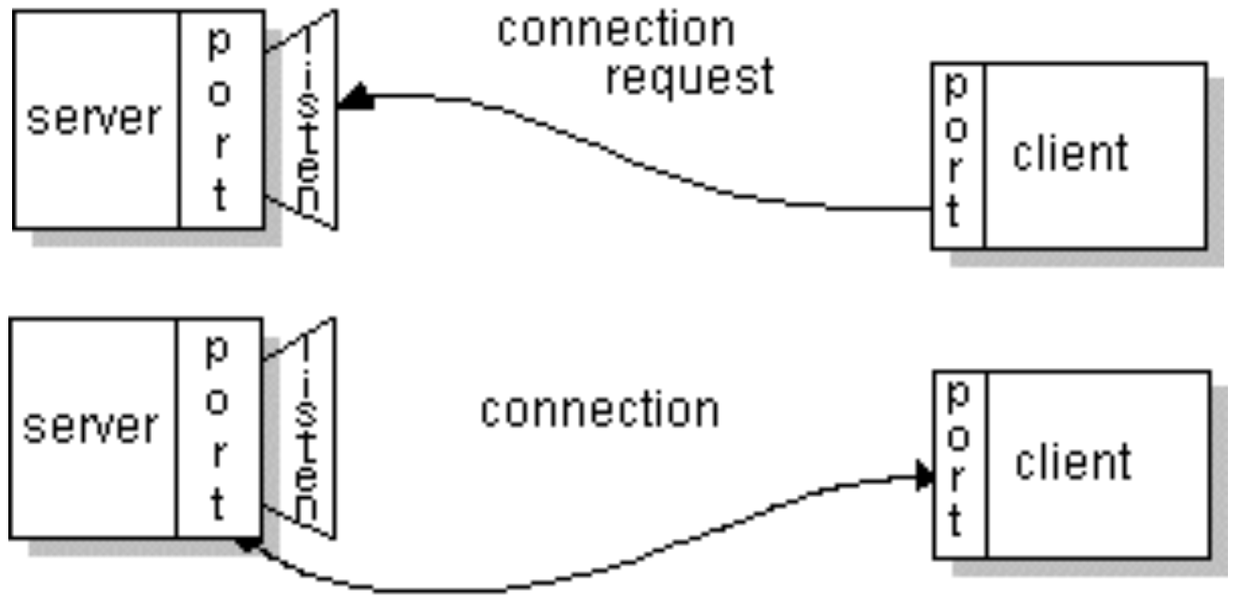
Sockets

Java provides two socket objects:

- java.net.Socket - which handles the client side connection

- java.net.ServerSocket - which handles the server side connection of the socket.

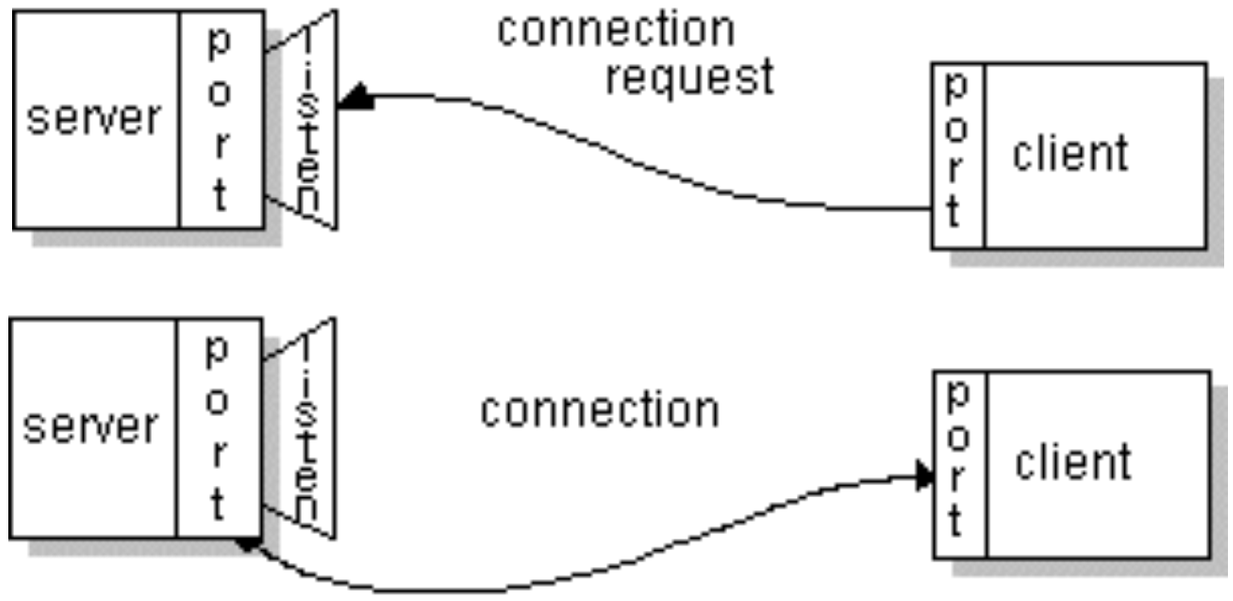
Sockets



The ServerSocket is *bound* to a port, which listens for incoming connections from clients.

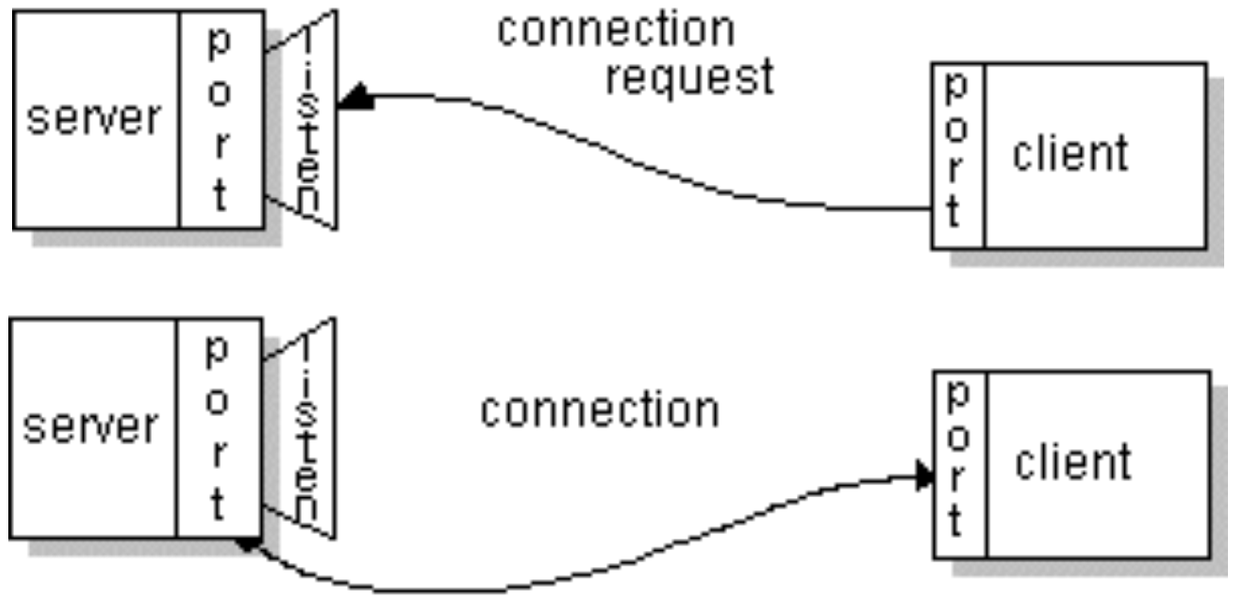
The client sockets connect to a hostname (like an ip address) and the port the ServerSocket is listening on.

Socket Definition



A *socket* is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

Sockets



Java's Socket classes sit on top of native code, hiding system details from your Java program. This is how compiled Java clients and servers on *heterogeneous* systems (systems with different architectures and operating systems) can still work together.

Web Connections

For connecting to the web (web pages) it is probably more appropriate to use Java's URLConnection and URLEncoder classes (more on that later):

http://docs.oracle.com/javase/tutorial/networking/urls/index.html

Reading and Writing from a Socket

See SimpleSocketDemo.tar|zip

Compile & Run

- Unarchive and change to directory that contains the src/ folder
- Compile
  - `javac -d build src/*.java`
- Run the server in one terminal window
  - `java -cp build SSServer 4444`
- Run the client in another terminal window
  - `java -cp build SSClient localhost 4444`