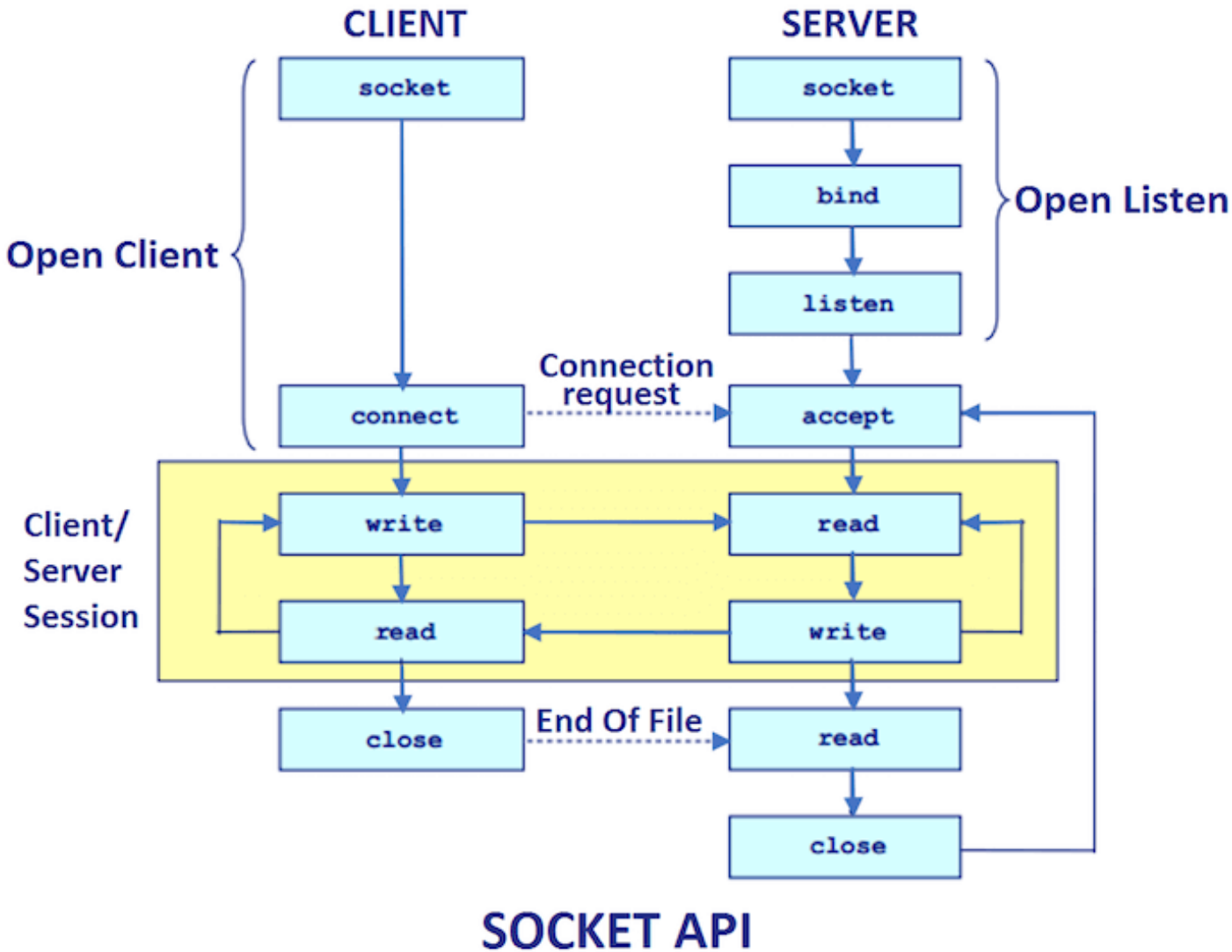
Socket Programming

YAO ZHAO


CLIENT
SERVER
socket
socket
bind
listen
Open Client
Open Listen
Connection request
connect
accept
Client/ Server Session
write
read
read
write
close
End Of File
read
close

SOCKET API

Java Socket

Class **Socket** and **ServerSocket** are in package **java.net**. The **Socket** class is used to communicate client and server. Through this class, we can read and write message. The **ServerSocket** class is used at server-side. The *accept()* method of **ServerSocket** class blocks the console until the client is connected. After the successful connection of client, it returns the instance of Socket at server-side.

Server-side

- To create the server application, we need to create the instance of ServerSocket class. Here, we are using 6666 port number for the communication between the client and server. You may also choose any other port number.
- The accept() method waits for the client. If clients connects with the given port number, it returns an instance of Socket.

```
serverSocket = new ServerSocket(6666);
```

```
Socket s = serverSocket.accept();
```

Please notice: If there is no client to connects the server, the entire program will remain blocked. Once a client connects the server, this code will return a socket instance to Socket s.

Client-side

- To create the client application, we need to create the instance of Socket class. Here, we need to pass the IP address or hostname of the Server and a port number. If server is running on same system with the client, we can use "localhost" as the hostname.

```
Socket sock = new Socket(hostName, portNumber)
```

Why need a port number?

- An Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.
- A device can run multiple applications at the same time, such as Email, HTTP, FTP, etc. The question then is, when a device receives a packet from the network, how does it know which application on the device the packet is sent to?
- Port: Just a number representing which application to run on a server.

Port number can be from 0 to 65535(2^{16}). One port number vs one application.

- Specific port numbers are reserved to identify specific services so that an arriving packet can be easily forwarded to a running application. For this purpose, port numbers lower than 1024 identify the historically most commonly used services and are called the well-known port numbers.

  Some well-known port numbers:

  ftp：23，telnet：23，smtp：25，dns：53，http：80，https：443

- Higher-numbered ports are available for general use by applications and are known as ephemeral ports.

Reading from and Writing to a Socket

```
PrintWriter out = new PrintWriter(sock.getOutputStream(), true);
BufferedReader in = new BufferedReader(new
InputStreamReader(sock.getInputStream()));


//send a message
String info = “xxxxx”;
out.println(info);
//don’t need flush if you set the autoFlush = true, otherwise you should use
//out.flush()



//receive a message
String fromServer = in.readLine();
```

Sample Code

- ClientDemo.java
- ServerDemo.java