

① Sockets

- What?
 - Types
 - Lifecycle
-

① Port Scanner

② Echo server

- Single connection
- multi connection
- multi threaded

la pil

- http-client

~~re~~ fetch

- axios

- requests

• post ()
1.

↓
HT TP

Post

• get

→ Socket

→ print some data to the
console

→ System.out.

→ print()

→ File()

→ Input Stream Reader

import os ←
sys

Network



Socket

↳ software entity

↳ allows devs. to send /
receive data over the
network (IPC)

|
Inter
Process

Types of sockets

 — TCP — connection
— UDP — connectionless

① Stream sockets \rightarrow TCP — TCP header.
— connection oriented

② Datagram \rightarrow UDP
— connectionless \rightarrow UDP header

③ Raw?

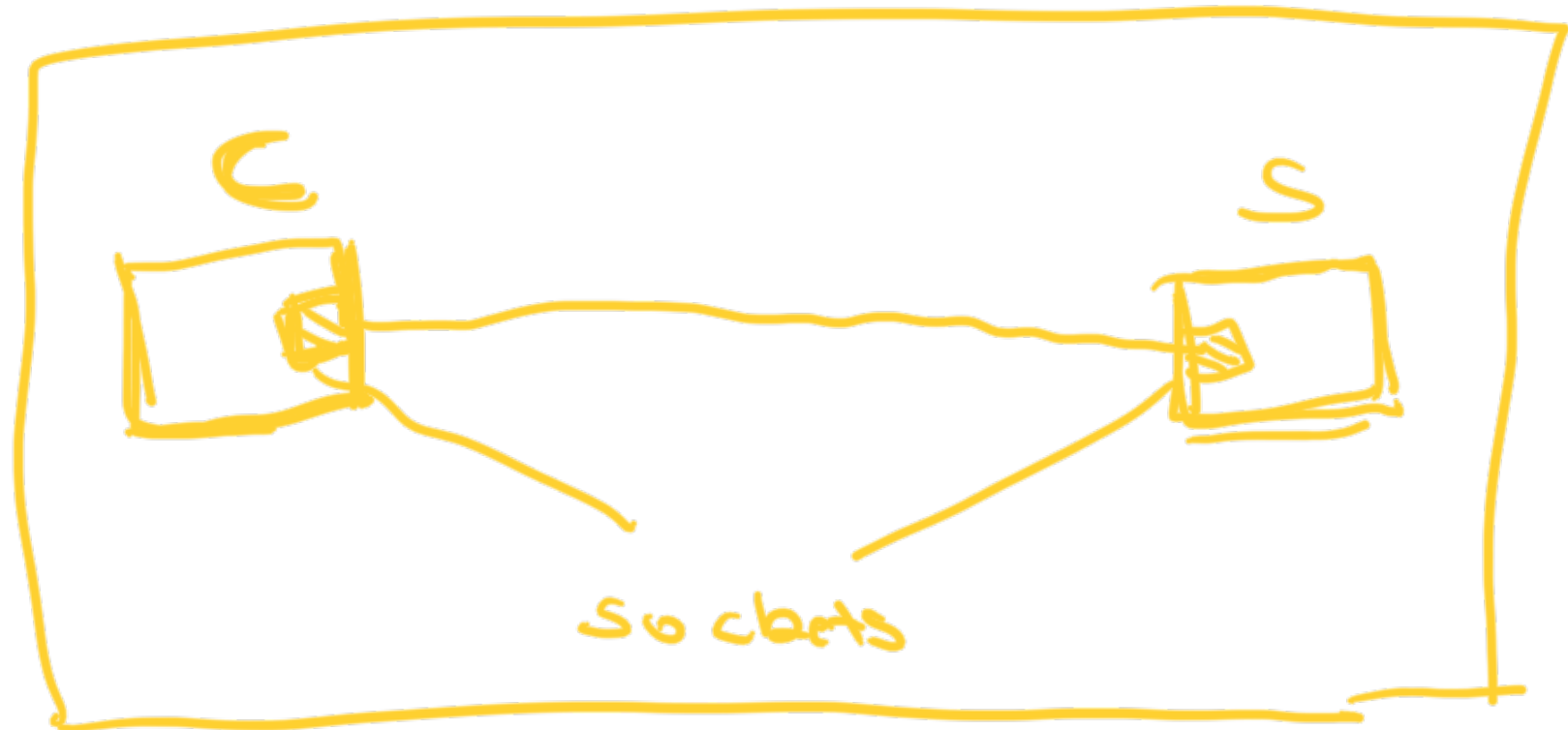
\rightarrow Testing

Socket

① Source

- IP

- PORT



② Destination

- IP

- PORT

□
|
IP
Port

□
IP
Port

Client



- ① Create a socket
new Socket()
- ② Create a connection
(IP, PORT)
[Remote]
- ③ Send data
- ④ Receive data
- ⑤ Close the socket

Server

- ① Create a socket
- ② Bind your
socket to a
port
- ③ Listening
- ④ Receive data
- ⑤ Send data
- ⑥ Close the socket





<u>Local</u>	[<u>Client IP</u>	,	<u>Client Port</u>)
<u>Server</u>	[<u>Server ID</u>	,	<u>Server Port</u>)



192.168.1.1

6001



S

① Socket - SI

② SI - 127.0.0.1
- 6000

③ Listening - 6000

Binding

c

① Socket S2

② Connection →

③ Send data

④ 'Receive data
⑤ Send data

④ Receive data

⑤ Close

⑥ Close

Req → Server



Ephemeral port

↓

Temporary

A diagram showing the text 'Ephemeral port' underlined. An arrow points down from the underline to the word 'Temporary'.

②

192.168.1.1

6291

localhost

6000

①

Port Scanner

- open ports
- close

- check if a port is open?



D o o d

Port scan

— all open ports

! ... end

→ connect (port)

closed

OPEN

4:46 → 4:50

10:20
