NETWORK PROTOCOL

msgs sent over wire or network ≡ msgs sent over the internet from 1 machine to another machine

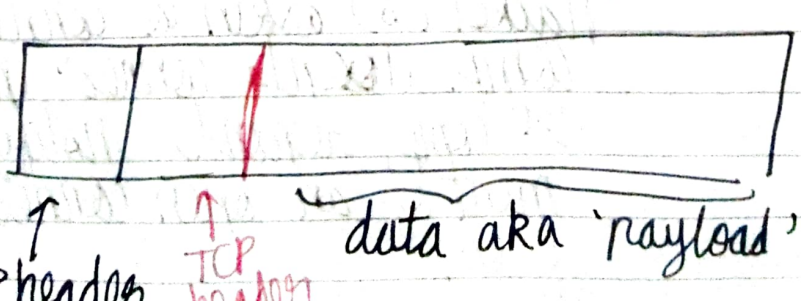
Network protocol consists of:
- types of msgs
- struc.
- order
- response to a msg, what shud it look like if present
- rules on when msgs can be sent to 1 another

3 main protocols:

1) IP → Internet Protocol

i) Modern internet effectively runs on IP. When 1 machine (client) interacts with another (server) and sends data to it the data is sent in form of IP packet (fundamental unit bldg blocks of data sent b/w machines i.e. communication)

ii) IP packet → made up of bytes



iii) IP header - At beginning of packet. Contains imp info → source IP address, destination IP address (machine sending data) (machine data will go to)

```
const httpResponse = {
    statusCode: 200,
    headers:
        {
        'access-control-allow-origin': 'https://www.goog
        'content-type': 'application/json',
        },
    body: '{}'
}
```

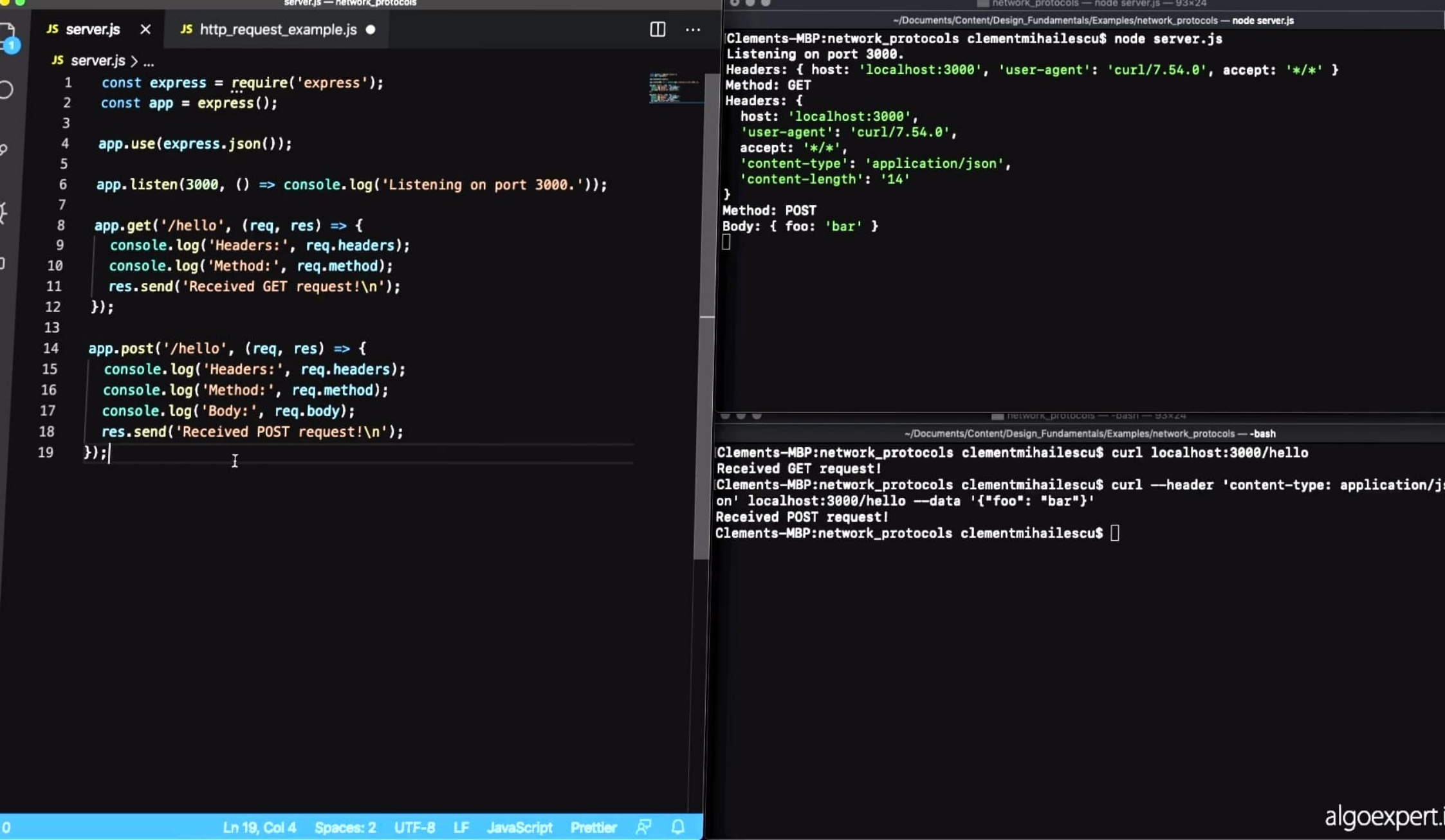
Path $\xrightarrow{\text{contains}}$ logic $\xrightarrow{\text{gets executed}}$ as per path provided in req

Headers → collection of k-v pairs containing imp. meta data or info abt req.

eg: 403 status code → data you're req. in is forbidden

NOTE status code → describes type of response status codes are like guidelines that you can alter as per requirement

eg:- 404 status code = requested piece of data not found.


server.js — network_protocols
server.js
http_request_example.js
server.js > ...
1 const express = require('express');
2 const app = express();
3
4 app.use(express.json());
5
6 app.listen(3000, () => console.log('Listening on port 3000.'));
7
8 app.get('/hello', (req, res) => {
9 console.log('Headers:', req.headers);
10 console.log('Method:', req.method);
11 res.send('Received GET request!\n');
12 });
13
14 app.post('/hello', (req, res) => {
15 console.log('Headers:', req.headers);
16 console.log('Method:', req.method);
17 console.log('Body:', req.body);
18 res.send('Received POST request!\n');
19 });
Ln 19, Col 4
Spaces: 2
UTF-8
LF
JavaScript
Prettier
network_protocols — node server.js — 93×24
~/Documents/Content/Design_Fundamentals/Examples/network_protocols — node server.js
Clements-MBP:network_protocols clementmihailescu$ node server.js
Listening on port 3000.
Headers: { host: 'localhost:3000', 'user-agent': 'curl/7.54.0', accept: '*/*' }
Method: GET
Headers: {
host: 'localhost:3000',
'user-agent': 'curl/7.54.0',
accept: '*/*',
'content-type': 'application/json',
'content-length': '14'
}
Method: POST
Body: { foo: 'bar' }
network_protocols — -bash — 93×24
~/Documents/Content/Design_Fundamentals/Examples/network_protocols — -bash
Clements-MBP:network_protocols clementmihailescu$ curl localhost:3000/hello
Received GET request!
Clements-MBP:network_protocols clementmihailescu$ curl --header 'content-type: application/j
on' localhost:3000/hello --data '{"foo": "bar"}'
Received POST request!
Clements-MBP:network_protocols clementmihailescu$
algoexpert.