Day-17 (nodejs - 3)

Sunday, 11 May 2025　5:10 PM

HTTP methods

GET → fetching data (retrieve data)

POST → send data (submit data / change on server)

PUT → updation (complete) ⎫ Better coding prac.

PATCH → updation (partial) ⎭

DELETE → removal

everything → put

patch

Patch

Patch
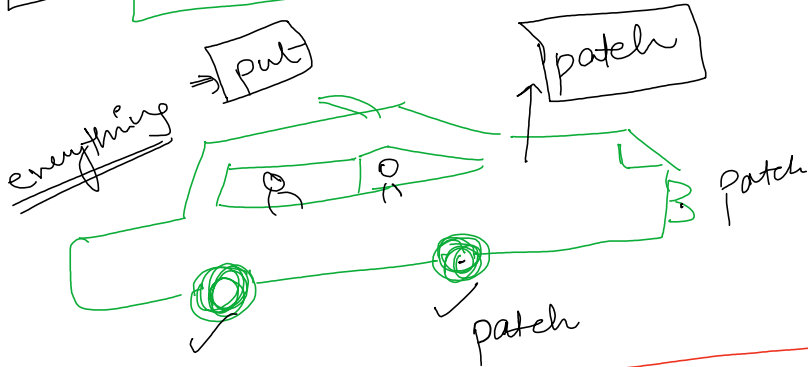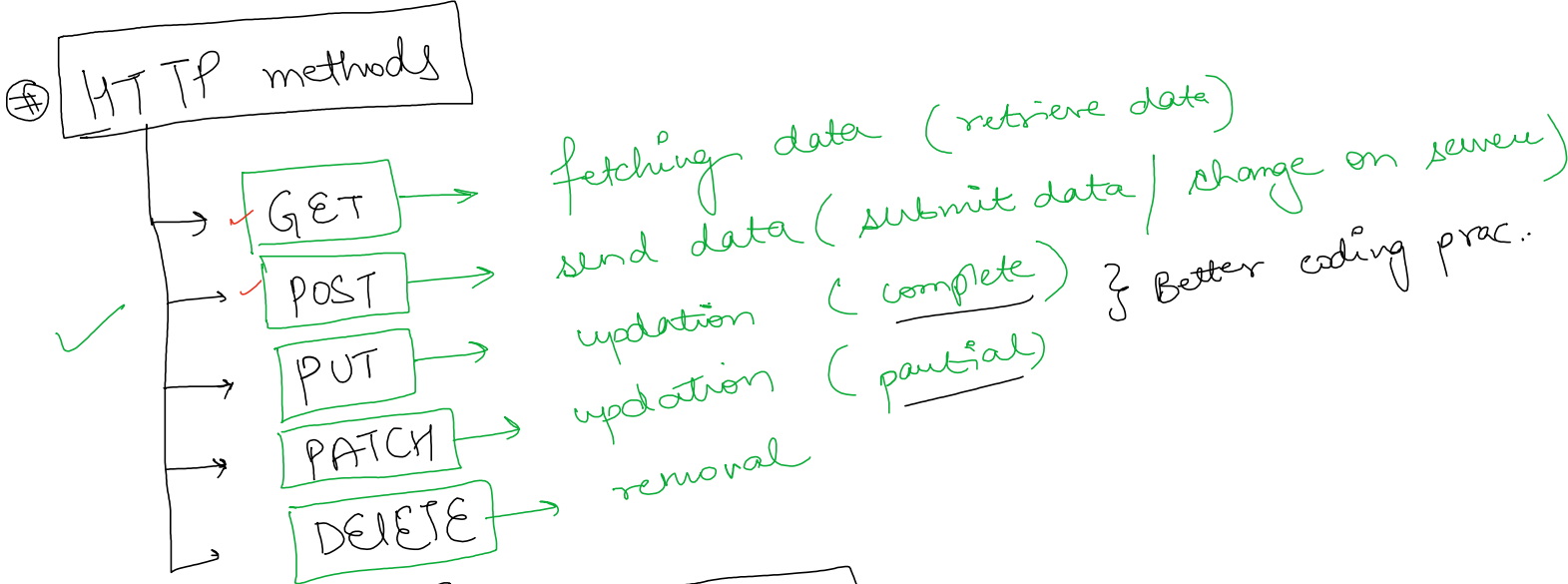
By default your browser hits which req? — GET
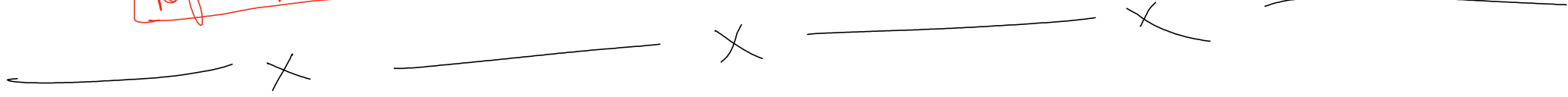
Express "framework"
↳ create servers

app.get

app.post ⎫ How?
put ⎭

[POSTMAN ✓
  THUNCLIENT ✓]

\# REST Api → Architecture → way of writing code

Representation
State
Transfer

Application

Blog application

↓

Apis

↓

C R U D

app.post (' / maikuch bhi banaou ', ()⇒{})

→ path

fellow developers

not to be remembered

general knowl.
- \# Restful routing → page create
- * Rest api → data exchange
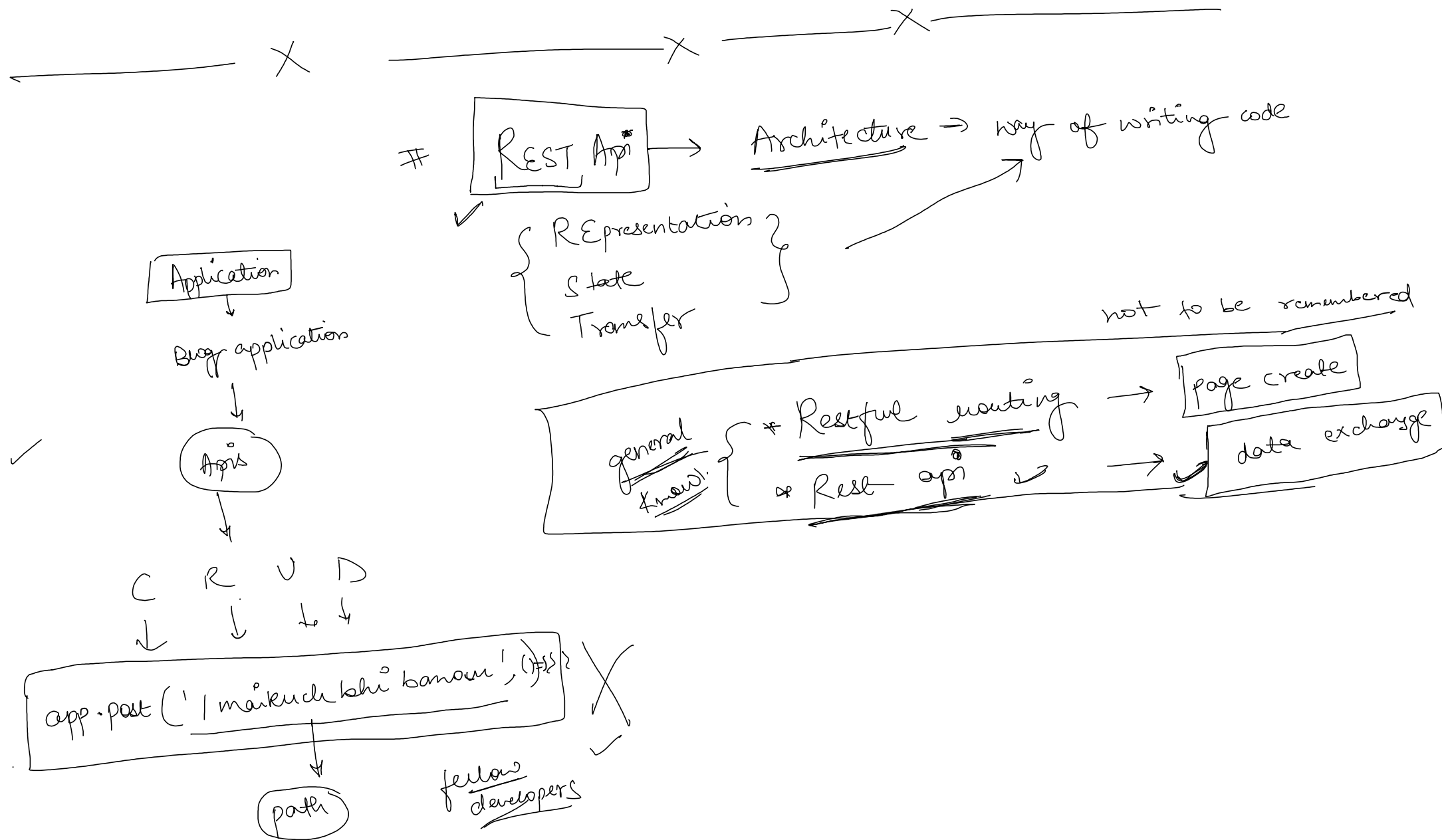
Me → Restful table

L ⟵ R   Blog

Route name	URL ③	HTTP Verb ②	Description ①
Index	/blogs	GET	Display a list of all blogs
New	/blog/new	GET	Show form to make new blogs
Create	/blogs	POST	Add new blog to database, then redirect

methods

① app.get (' / home ' , --)

② app.post (' / home ' , --)

Show	/blogs/:id	GET	Show info about one blogs
Edit	/blogs/:id/edit	GET	Show edit form of one blog
Update	/blogs/:id	PUT / PATCH	Update a particular blog, then redirect
Destroy	/blogs/:id	DELETE	Delete a particular blog, then redirect

∞ routing
or
dynamic ""

: id

url

routing or dynamic "

req. params

2
3
4

∞ apis

http: // localhost : 8000 / blogs

```
2
3
4
7
:id
```

req. params

100

1
2
3
4
5
:

———— X ———— X ————————

NodeJS → RUNTIME ENV.

→ JS without Browser
→ server
→ CLI
→ native apps

node + enter ⟹ REPL

→ window X   global ✓

BROWSER

→ global object = window
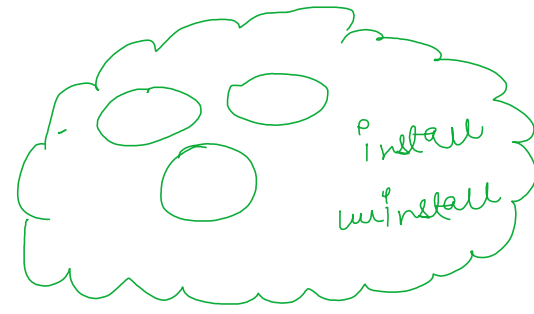
NODE JS

global object = global

system ✓

→ file system X | file system...

→ document ✓ | document X

install
uninstall

npm → node package manager

node index.js ✓ } Run
or
node index ✓

npm init or npm init -y   install

↳ package.json   ↑ "nodemon index.js"

↳ "start" : "nodemon index.js"

index.js   math.js

require (common JS) → fn1 → module . e

import (module)   fn2
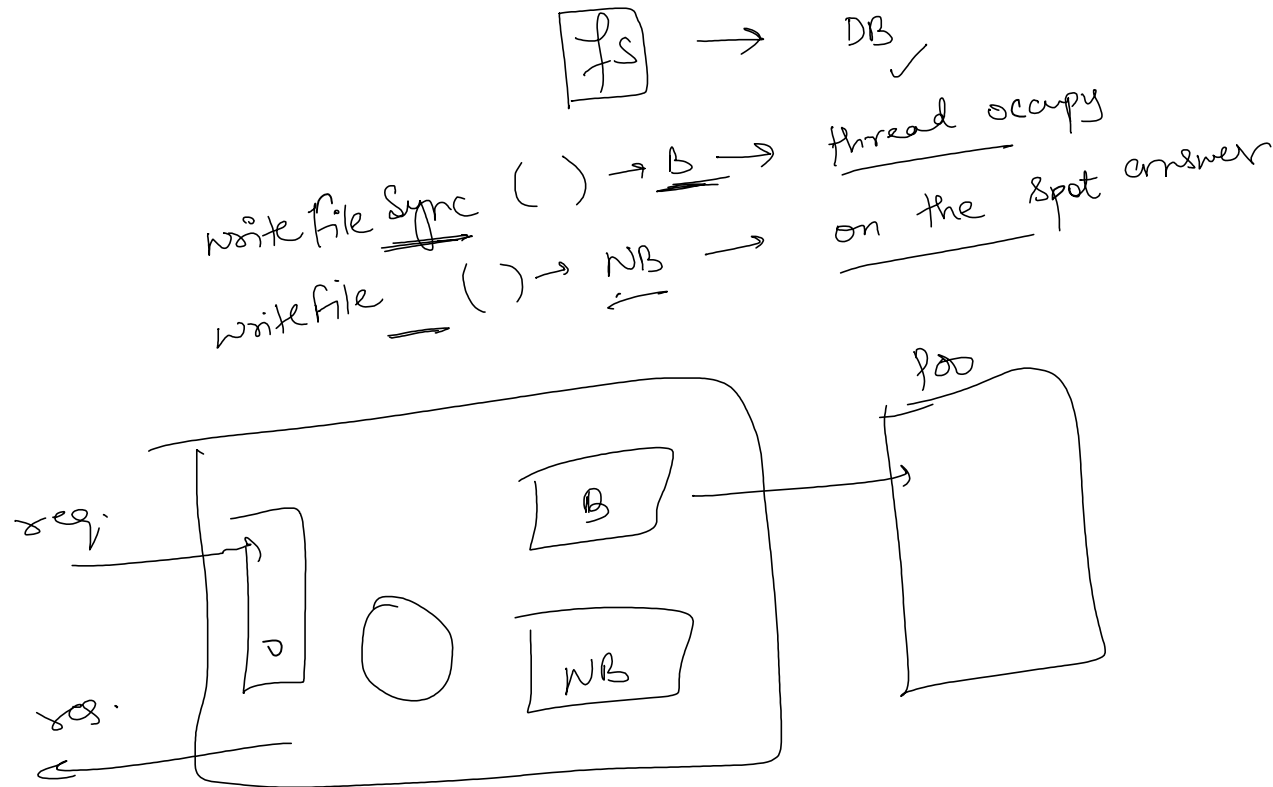
index.js   math.js

const { s1, s2 } = require ("./math")   export or export default s2

                   ↑ function sam ( ) {

                        clg ("Balle Balle")

$s_2()$

import { s_1 }, s_2 from "math. js"

module . exports = sam ;

" " = { sam 1 , sam 2 2 } ;

--- ✗ --- ✗ ---

fs → DB

write file Sync ( ) → B → thread occupy

write file ___ ( ) → NB → on the Spot answer

req.

B

D O

WB

res.

foo

require (' fs ')

| .) math - js |
| " fs " " http " |

--- ✗ --- ✗ --- ✗

http → way / rule { http https ws }

req, res

http . Create Server ((n) ⇒ { })

↳ request listener

res.end()

req. params → :id path

req. query → ? query

req. body → POST (data)

req. headers → extra info

HTTP
 └ GET → retrieve
 └ POST → send
 └ PUT ─┐
 └ PATCH ┘ update ┌ complete
                  └ partial
 └ DELETE → removal

Express → framework (http)
  ↕
Nodejs
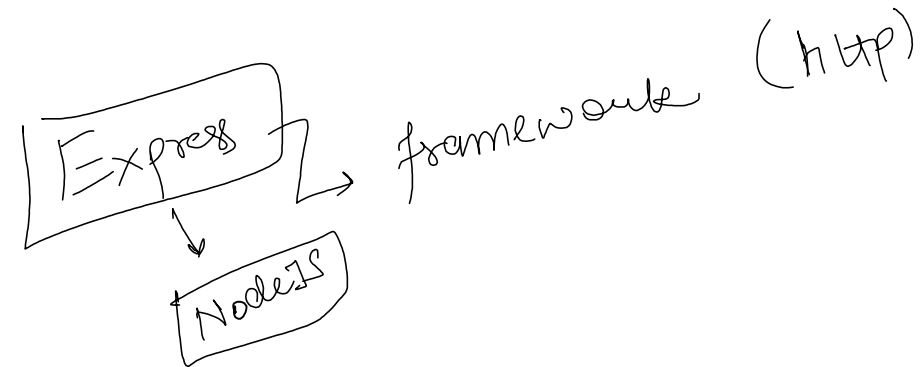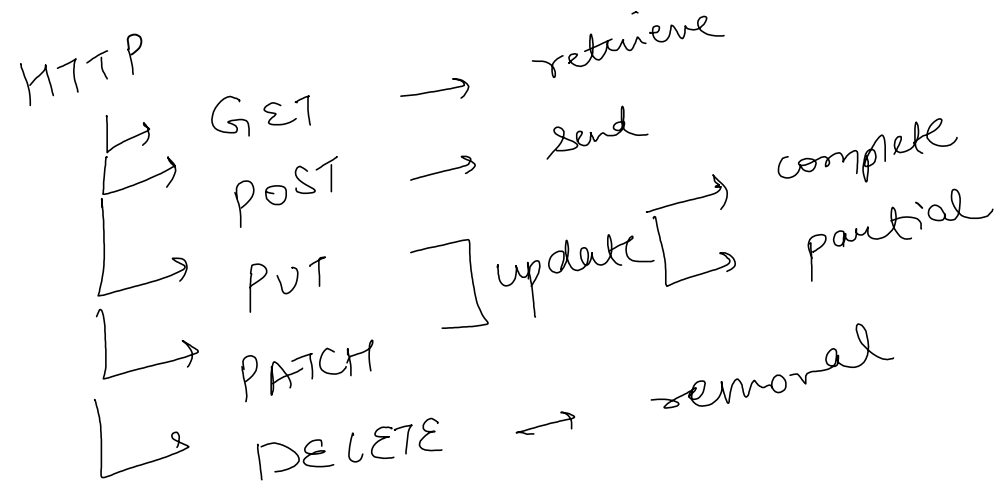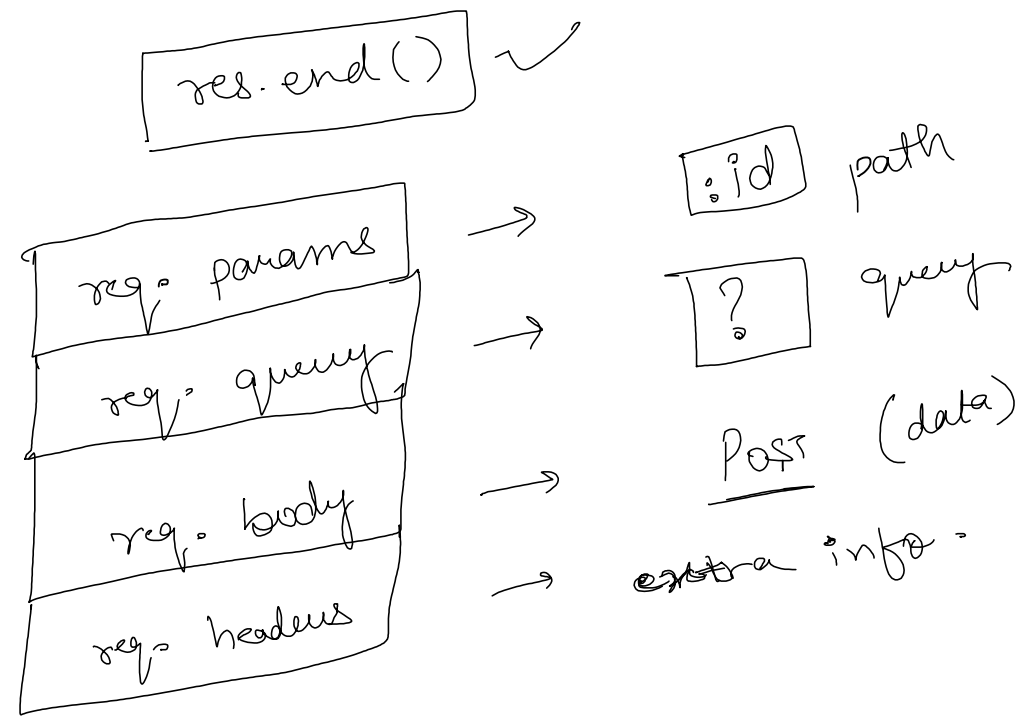
const express = require('express')
const app = express() // entire application
                req, res
                (.) => { }

app.get ("/home")

app.listen (8080, ( ) => { --- }

thunder client

→ POST
→ PUT
→ PATCH
→ DELETE

✓ REST → Way of writing code

→ common ground

Method + path ⇒ Task