Service : classe qui permet la liaison entre la partie FE et BE.

Envoyer des Requests et faire Retourner des Responses

HttpClient (بوسطجي)

↓

module

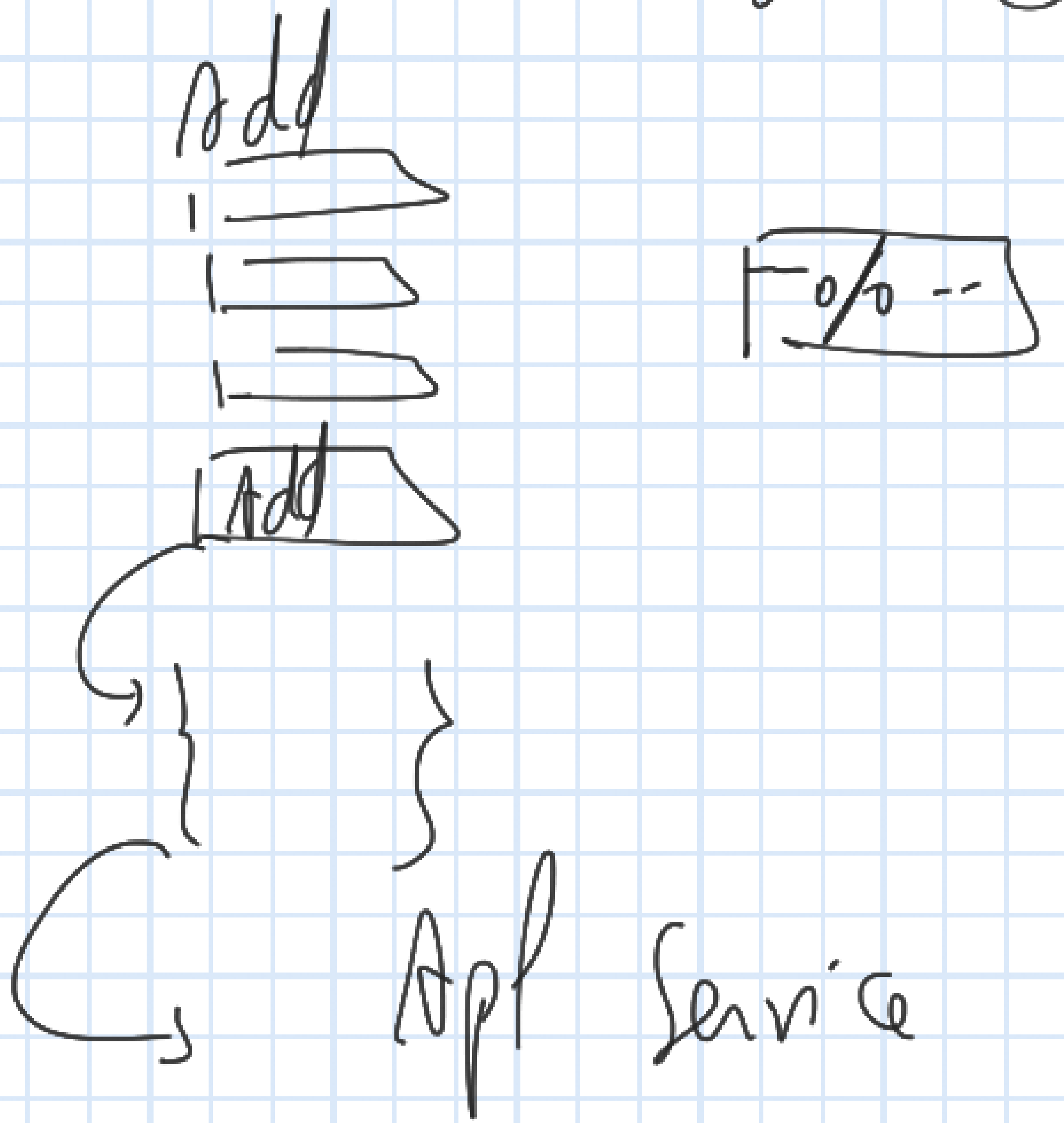
importer HttpClientModule ds le app.module.ts (imports)

$ ng g s N— Service

———.ts ———.spec.ts

1) déclarer HttpClient ds le constructor
2) — l'adresse de destination
3) — les méthodes

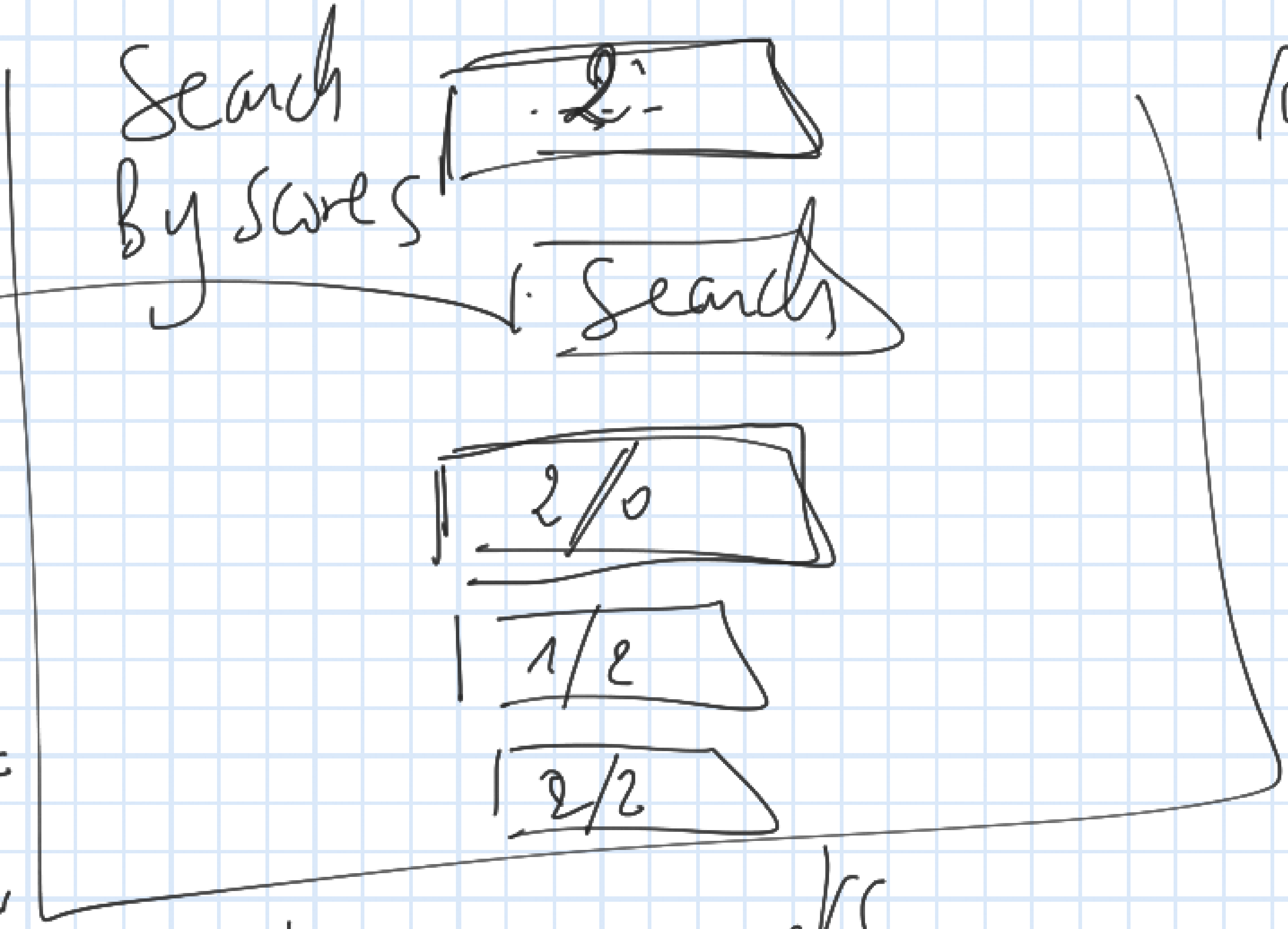
On génère un Service selon les modèles de la DB



match
team
stadium
player
user
⋮

Service

addMatch
getMatchById



Afficher / Récupérer les objets ayant un score ou [illegible] = 2

(click) = "Search()"

{ score : 2 }

— obs

search( ) {

// API service

}

// Get input value

// Get all objets

// Parcourir

if

Response

Service

{ score : 2 }

SearchMatches By Scores (obj) {

return httpClient. Post (@ , obj)

~~get~~

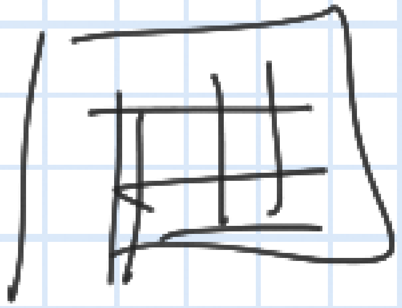
~~put~~

~~delete~~

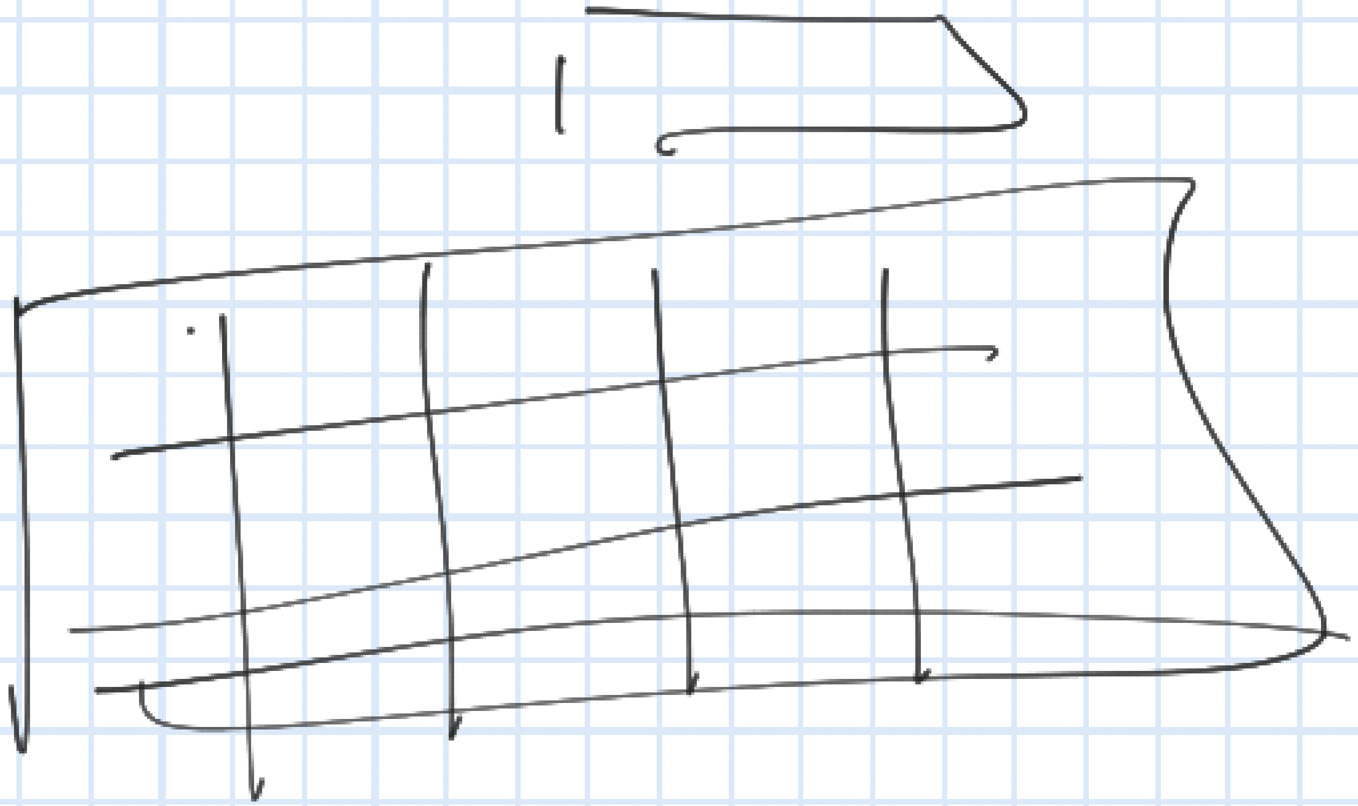
} [{ , { , { , {]

BE

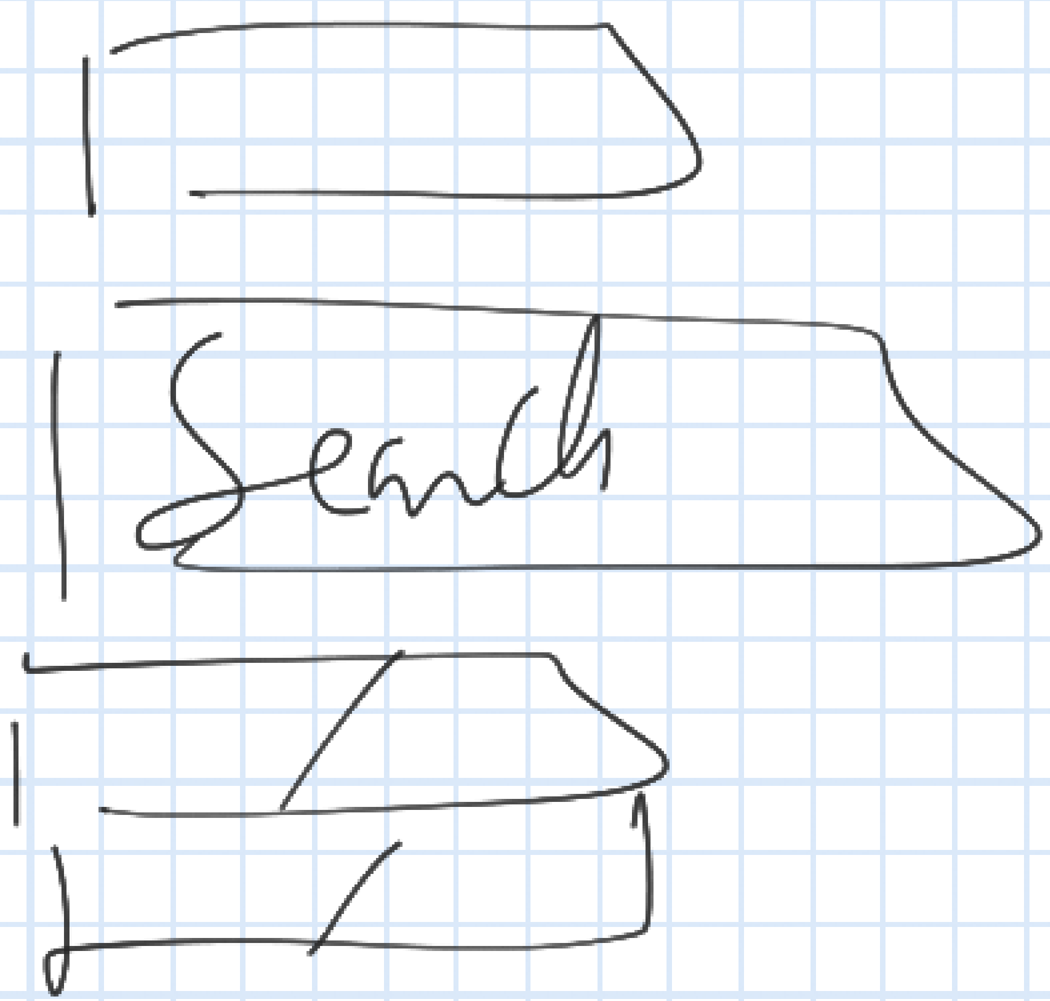
{ score : 2 }

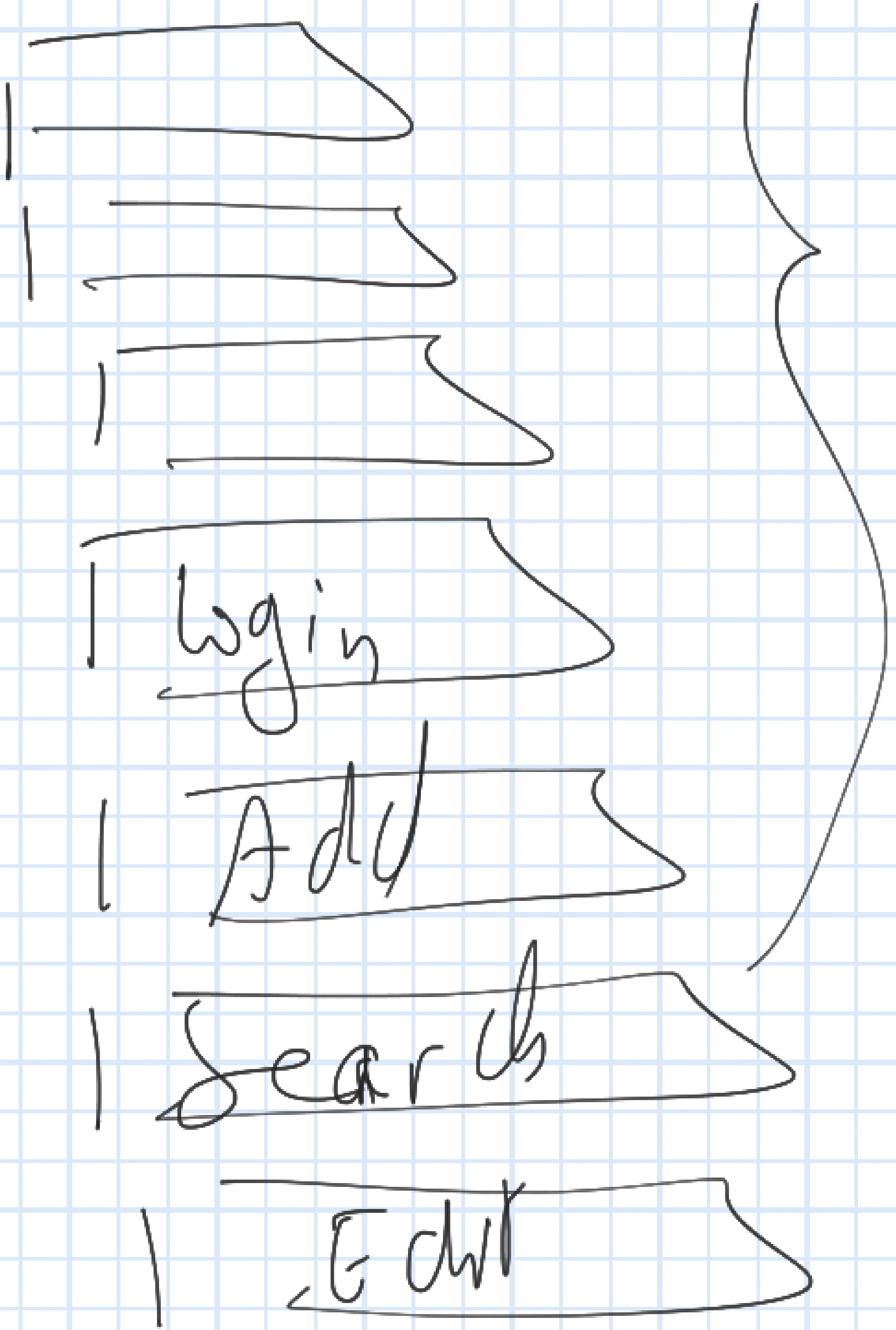


Filter



Search





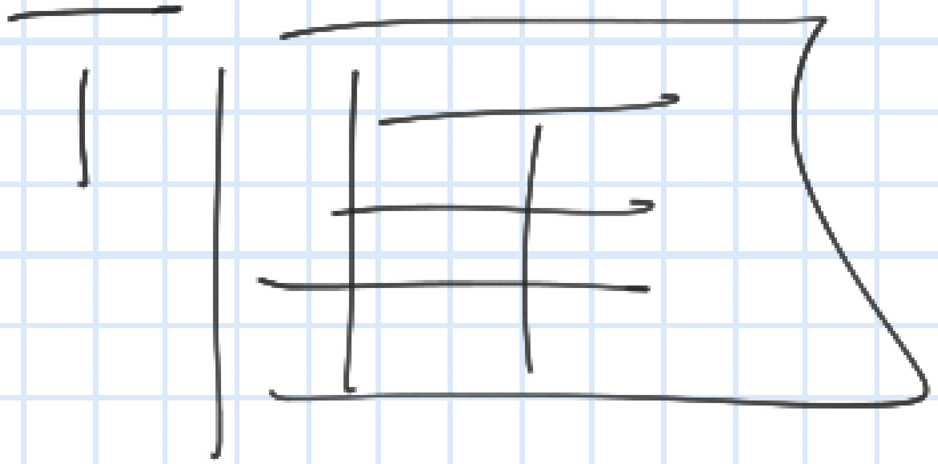
formulaire

{ }

Post

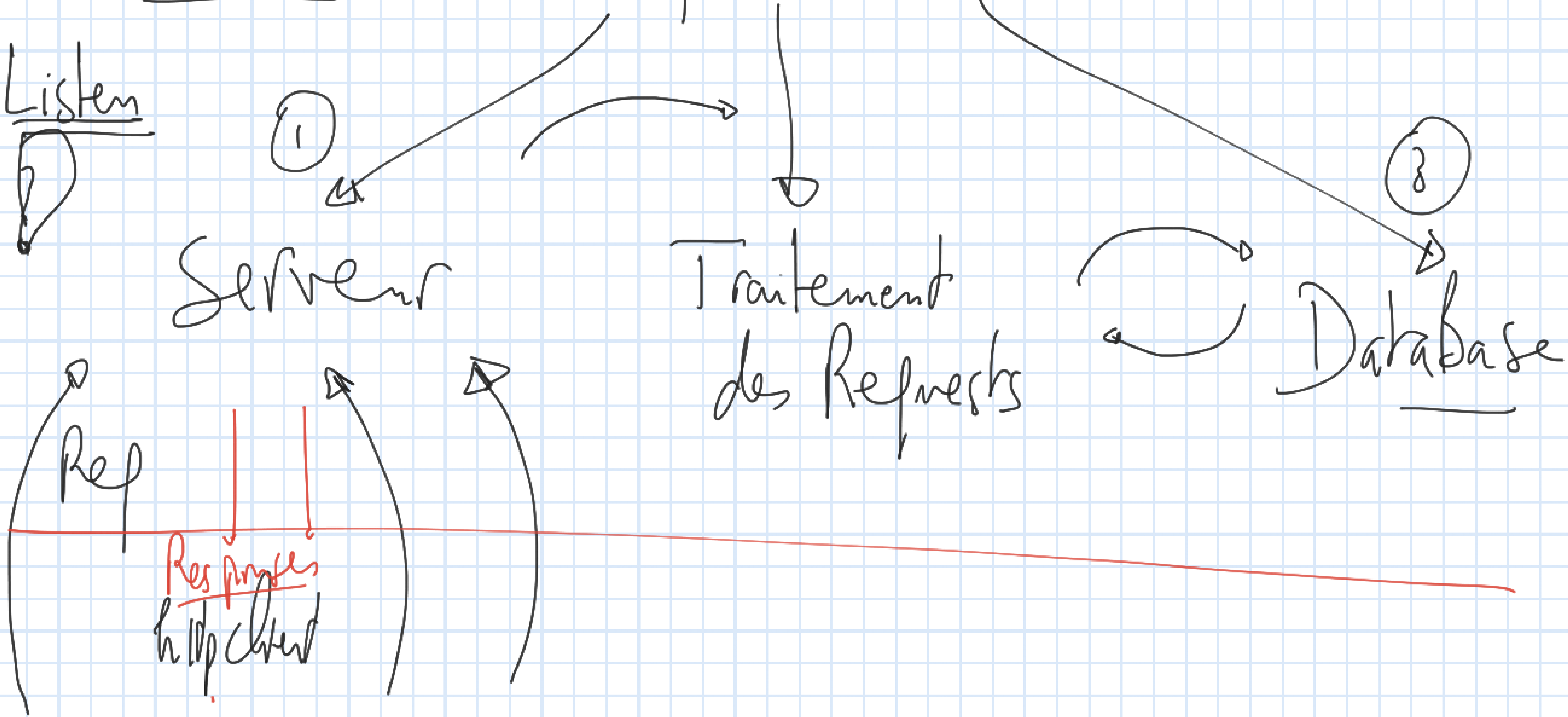
BE

```
{ score : 2 }

for (   T ) {
      if (  _ _ == 2 )

}
```



```
X = [ { }, { } ]
```

BE : la partie invisible au client



Express JS :

module Node

```
$ npm i express
```

Framework BE JS

① Node JS

Env d'exécut° du code JS en dehors du navigateur

(ensemble de modules)

1M

② Git Bash (terml pr exécuter les Cmds Unix)

→ touch
→ mkdir
clear

③ NPM

④
```
npm i -g @angular/cli
```

GitBash → VSCode

npmjs.com

1) npm i express

2) Structure du BE

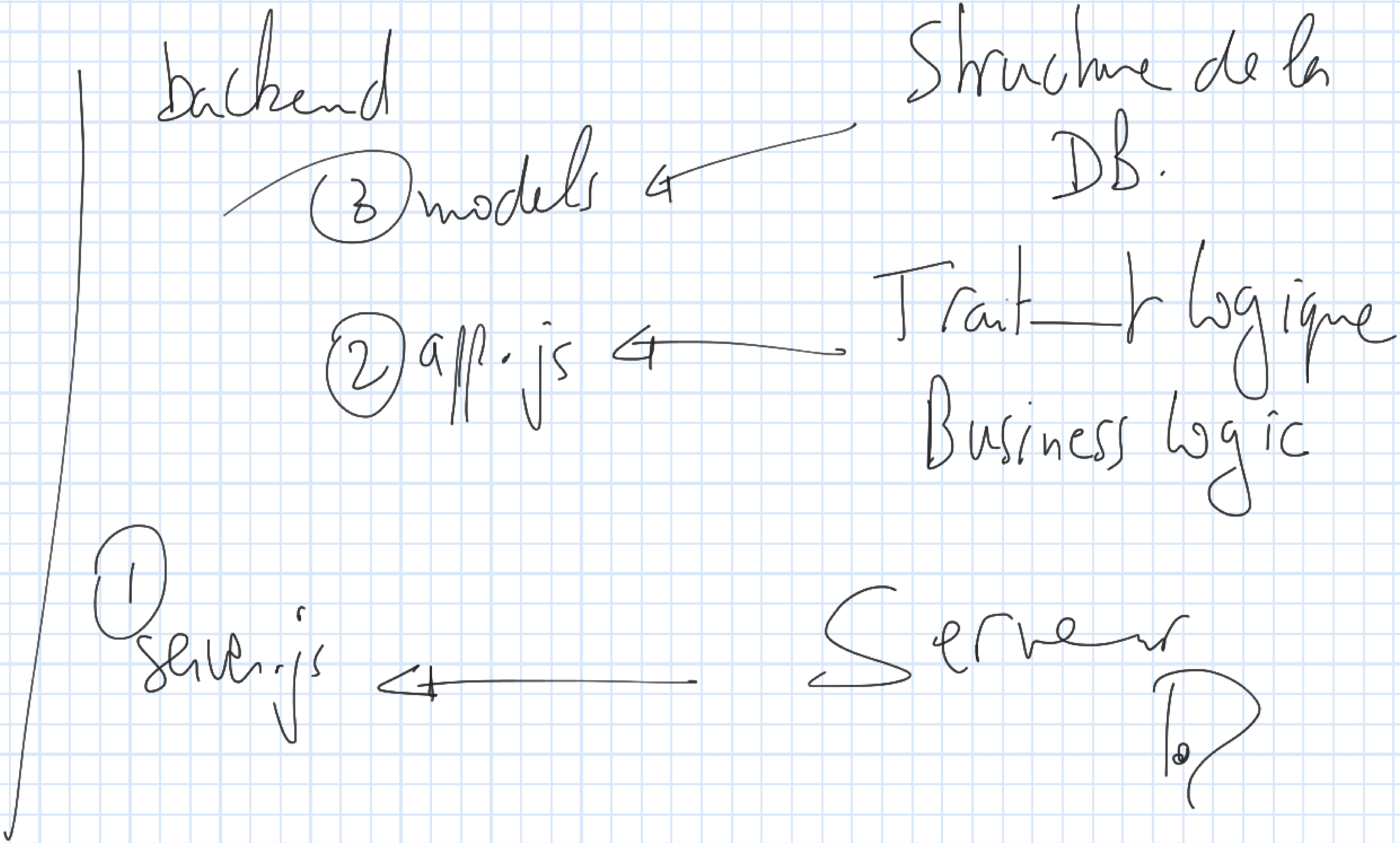
[illegible] un fichier

VSCode
- backend
  - models
  - app.js
- e2e
- node_modules
- src
  - app
    - controllers
    - pipes
    - services
- server.js

```
$ ng    new Pr—

      e2e
      n—
      src
        app
```

Créat° d' →

structure



backend

3 models ← Structure de la DB.

2 app.js ← Trait—t logique
Business logic

1 server.js ← Serveur

Po

$npm i body-parser

BE ⟹ JS

—.js

FE

- Angular: f— FE JS
- import ----
- ng new Project

BE

- Express: f— BE JS
- require(" No-Module ")
- app = express( )

Serveur BE :

app . listen(3000) → http:// l—host : 3000

↓

Applicat° BE

Req1 → Req2 → Req3 → http:// l— : 3000

→ @ serveur BE

Lancer le serveur FE

```
$ ng s -o
```

Lancer serveur BE

se pointer sur le projet sprint

$ ~~node server.js~~

```
$ nodemon server.js
$ npm i body-parser
```

BE → FE

↓

Responses

① JSON

Envoyer des réponses sous format JSON

FE → BE

{Post. } }
{Put.

② Récuperer des objets fro Request

① $ npm i body-parser

② Importer body-parser

③ Configurer body-parser

Player Service

addPlayer
editPlayer
deletePlayer
getAllPlayers
getPlayerById

- sepointer sur le dossier services
- ng g s user player

User Service

login( )
signup( )
editProfile( )