$ mongod

DB

$ nodemon server.js

BE

$ ng s -o

FE

$mongo

Gestión de la DB

$ Cmds

mongoose : module

assure la comm entre l'app et la DB

- connect(@server/DB/ N— DB)
- model,
- save
- schema
- find

Modèle

→ structure

- Type

)

match.js

→ schema

↓

ensemble d'attributs

- model (Name, schema)

↓

PascalCase

- module.exports = … ;

Importer le modèle ds app.js

1) app.post ("/matches", (req, res) => {

log (req.body) objet récupéré de la partie FE (formulaire)

let match = new Match ({ → le nom du modèle Type

attrModel : valObj

:

:

}) });

match.save ( )

Modèle → collection

Match   matches

User   users

```
app.get('/matches', (req, res) => {
matches → Match.find().then(
});
```

Donne le résultat de la recherche

méthode prédéfinie mongoose qui retourne l'ensemble de ts les docts

```
Match.find().then((data) => {
                  x
                  reponse
                  result  } => Le tableau d'objets récupérés de la Collection

      [{},{},{},{}]
  res.json({matches: data})
            ↓
            déclarée ds le Service
});
```

```
ret — httpClient.get<{matches: any}>
```

Compo———t :

Constr——r (private matchService : MatchService)

Récupérer la réponse du service →

this.matchService.getAllMatches().
subscribe((X) =>

this.matches = X.matches

Instance / Variable Globale déclarée dans le Comp——t

___.html

<tr *ngFor = "let m of matches"

B.L : getMatchByID.

http://l—:4200/matches/4

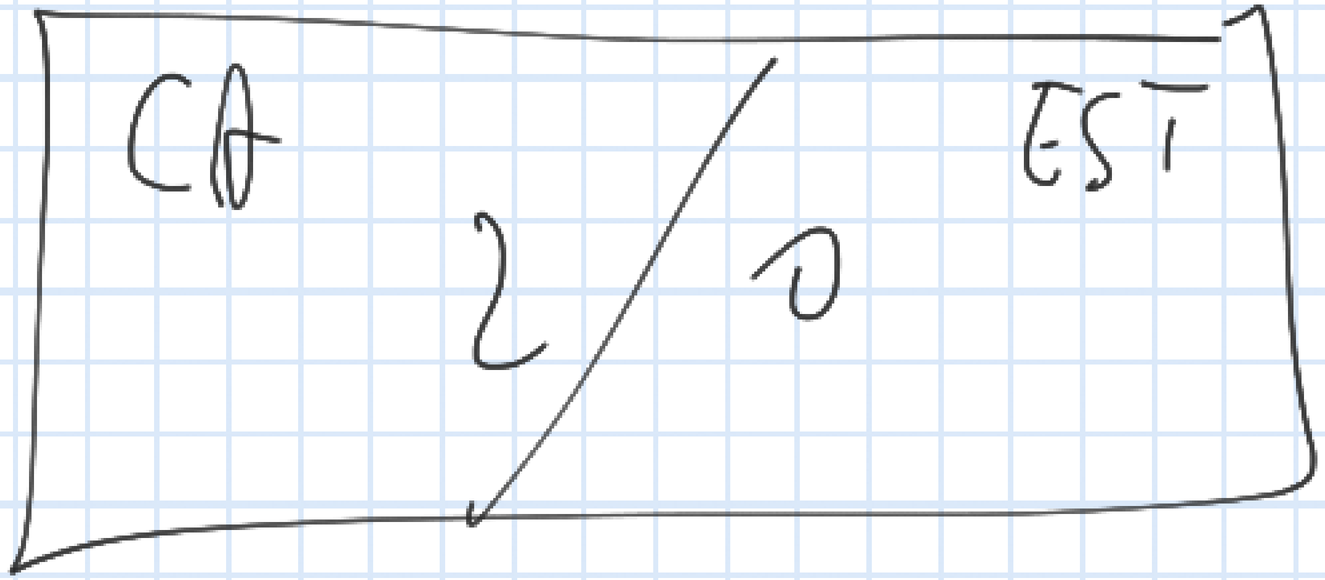
Compo—t : MatchInfoC—t

1) Get ID ← Path (Act—Route
(Snapshot))

2) Déclarat° du service

3) apl de la méthode
getMatchByID (this.id).
Subscribe (
)

MatchInfo



Service :

2
8
4

```
re___  httpClient.get( `http://l___:3000/matches/${id}` );
```

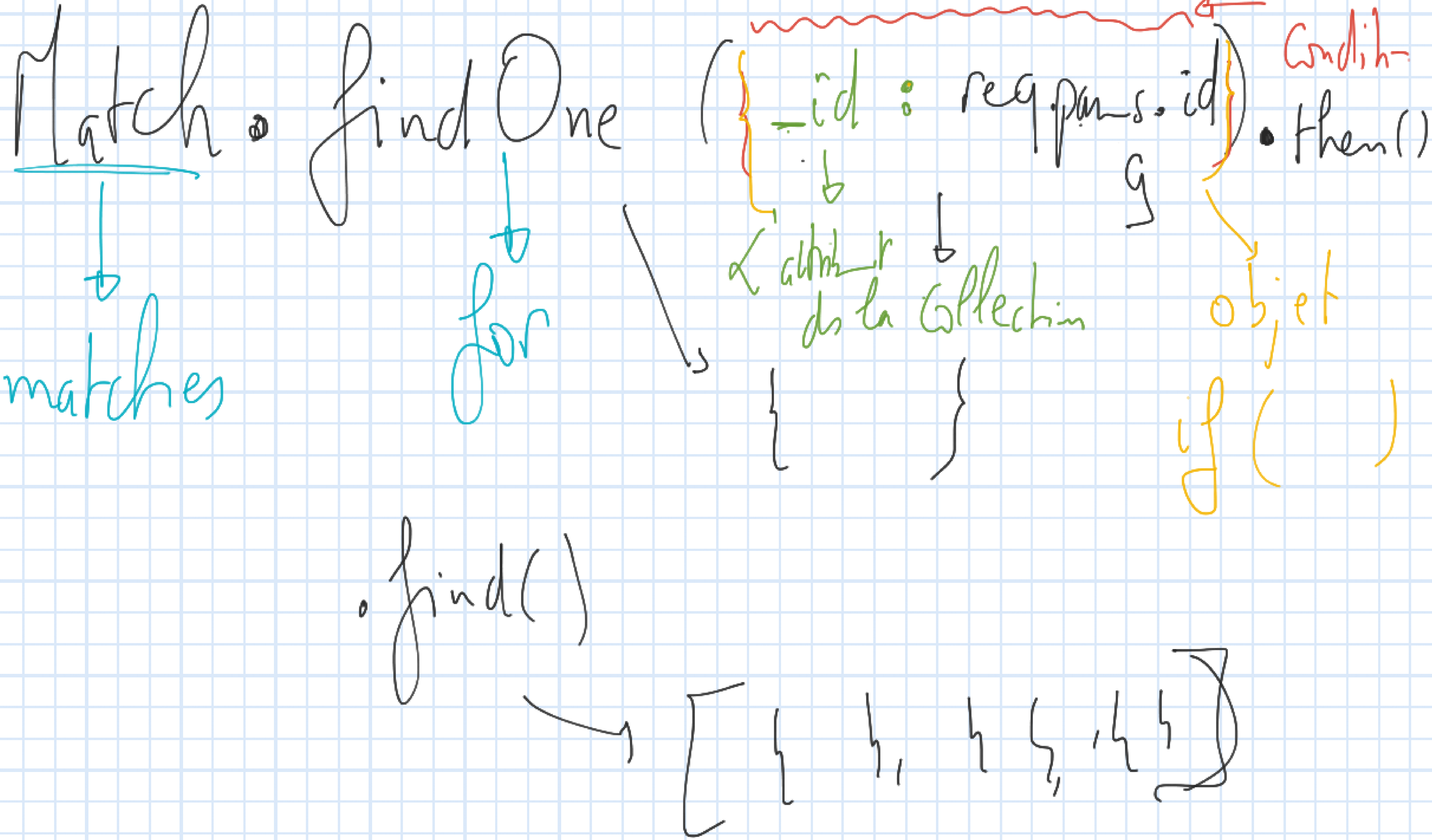
↳ app.js

```
app.get("/matches/:id", (req,res) => {
    log( req.params.id )    9
        // T. Logique
        //
})
```

```
$ show dbs

$ use N_DB
$ show collections
$ db.Coll-Na_.find( )
```

_id


Match . findOne ({_id : req.params.id}) . then()
Condition
matches
for
{ }
&lt; attribut ds la Collection
objet
if ( )
. find()
[{ }, { }, { }, { }]

```
instruct → .find     → [{},{},{}] .then(   )
         → .findOne  → { }  .then(   )
         → .findMany → [{},{}] .then(   )
         → .save

_.then( (data) => { //            })
```

data ↓ la réponse de l'instruction

Get All Players

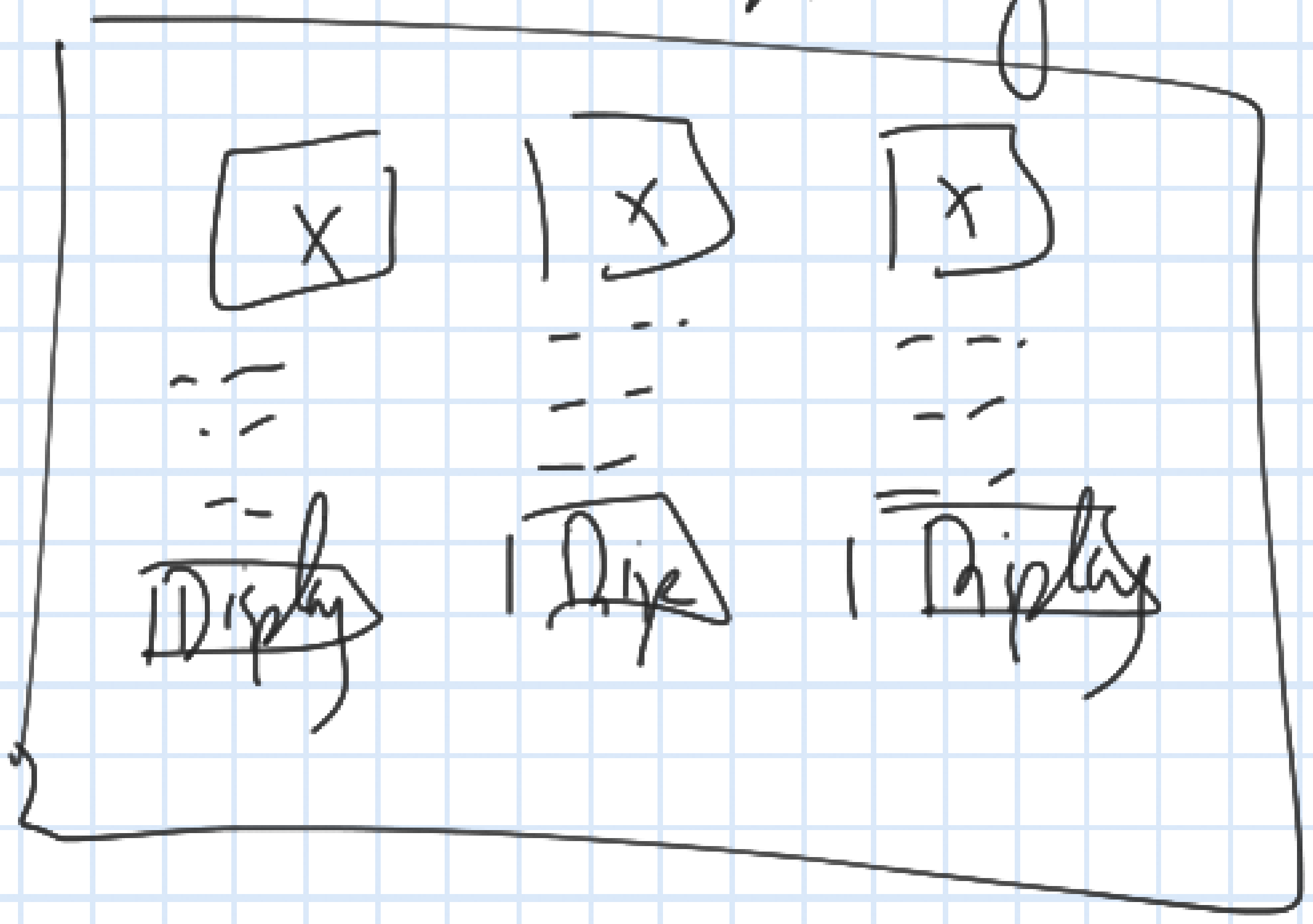
Get Player by ID

Resto :

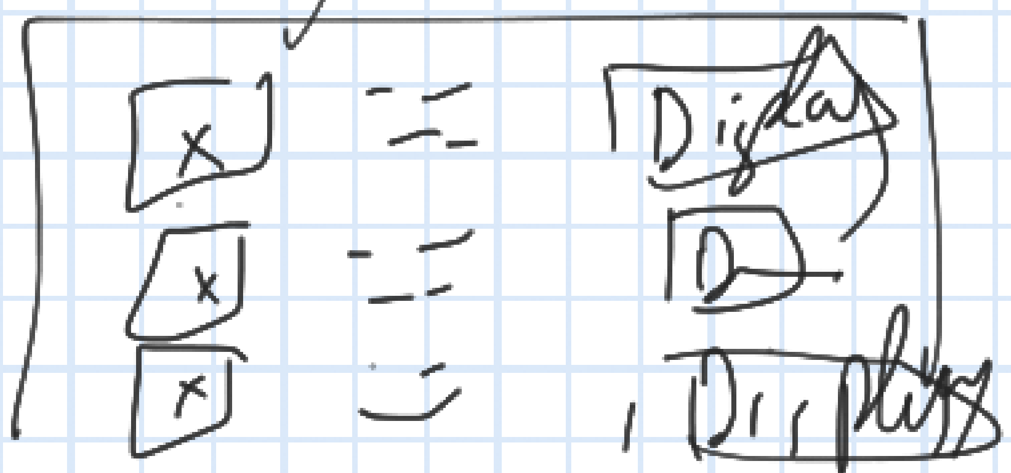
- addPlat → ↳ → form
- addChef → ↳ → form

Reactive . TDF

Display All Chefs → ... 4200 /Chefs



Display All Plats



+ Service + App.js +

Models DB

- add
- GetAll
- GetByID

(Click) Display

assure la navigat° vers un autre Compt plat-info, chef-info