

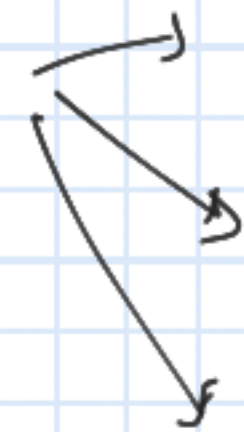
Add



this.info value / this object

TDF

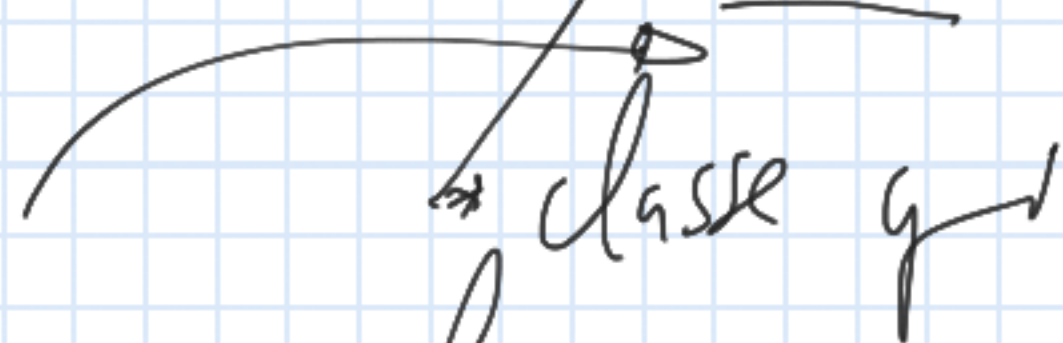
Rf



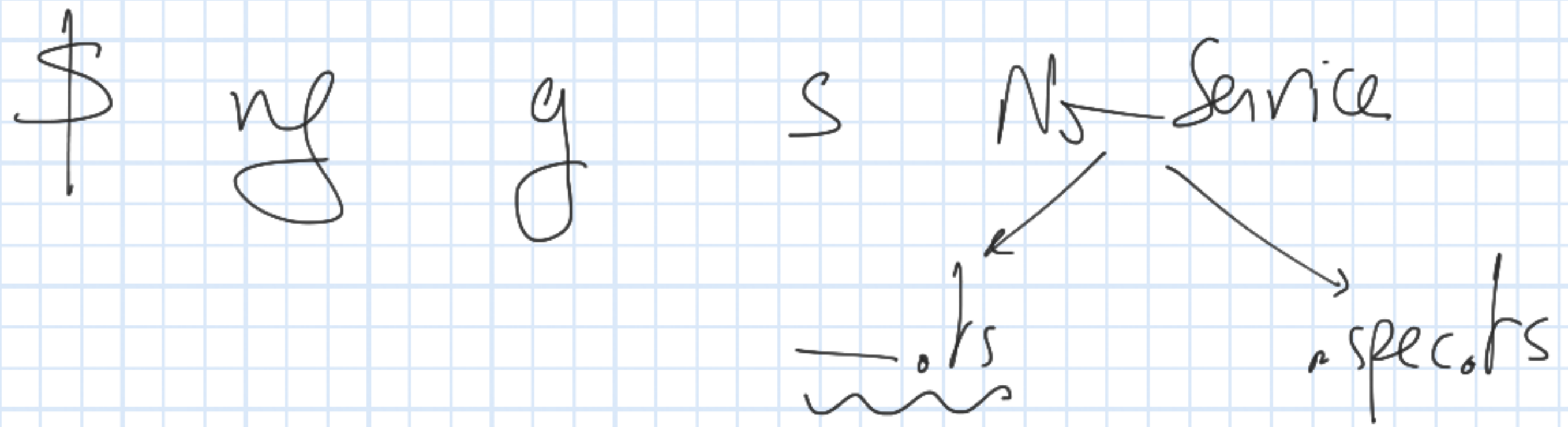
info -  
info model  
info submit

for Group  
for Group Name  
(click)

Service



assure la commun  
entre le RF et le BT



.) URL server BE

.) HttpClient

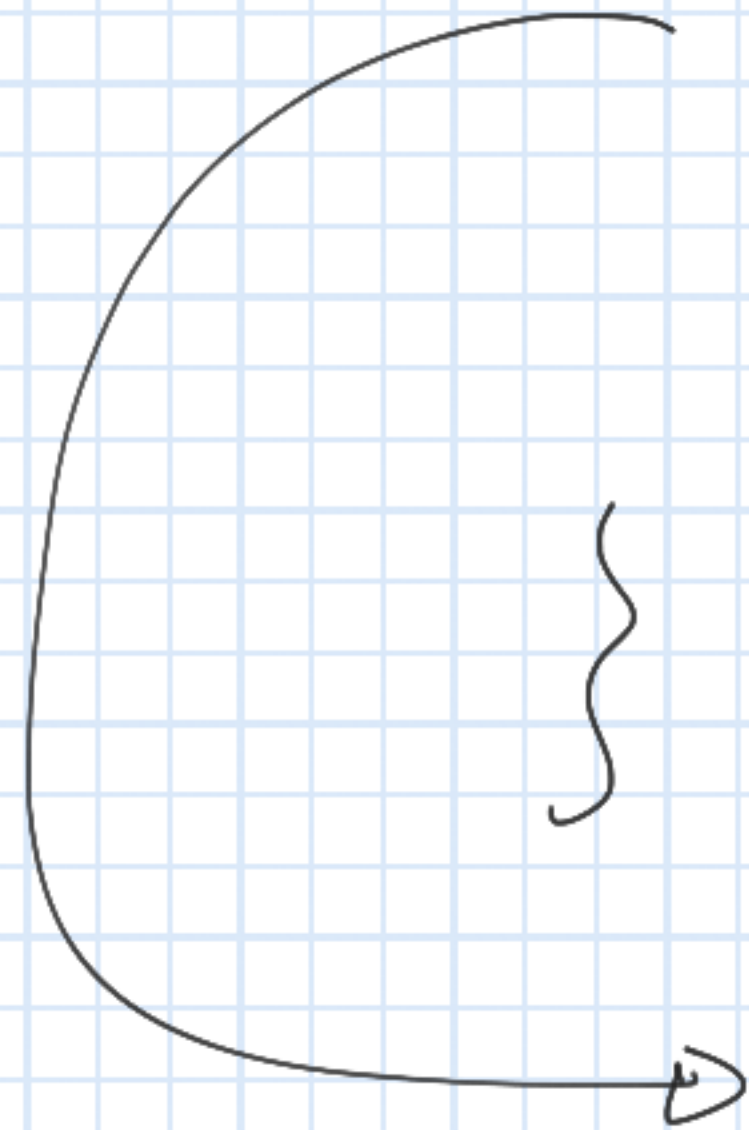
.) as methods

add Mark()

return HttpClient  
{ Action @SBE }

Add Match ( obj ) {

rel http.post ( @ , obj )  
    { put http: > 3000 / matches



server.js

http: > 3000  
→ listen ( 3000 )

→ Dispatch → app.js (ensemble de fonctions)

app.use(handleHTTP("path", (req, res) => {

// Traitement

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

req.body

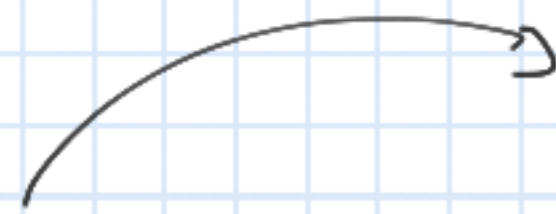
(Body Parser)

});

});



{ } { } { }



Service

httpClient.post

(@, obj1, obj2, obj3)

{obj1: { }, obj2: { },  
obj3: { } }

JSON.stringify

JSON.parse(req.body) ← [ { }, { }, { } ]  
res. {1, 1, 1}

app.js  
app.post("/matches", (req, res) => {  
 // let x = req.body;

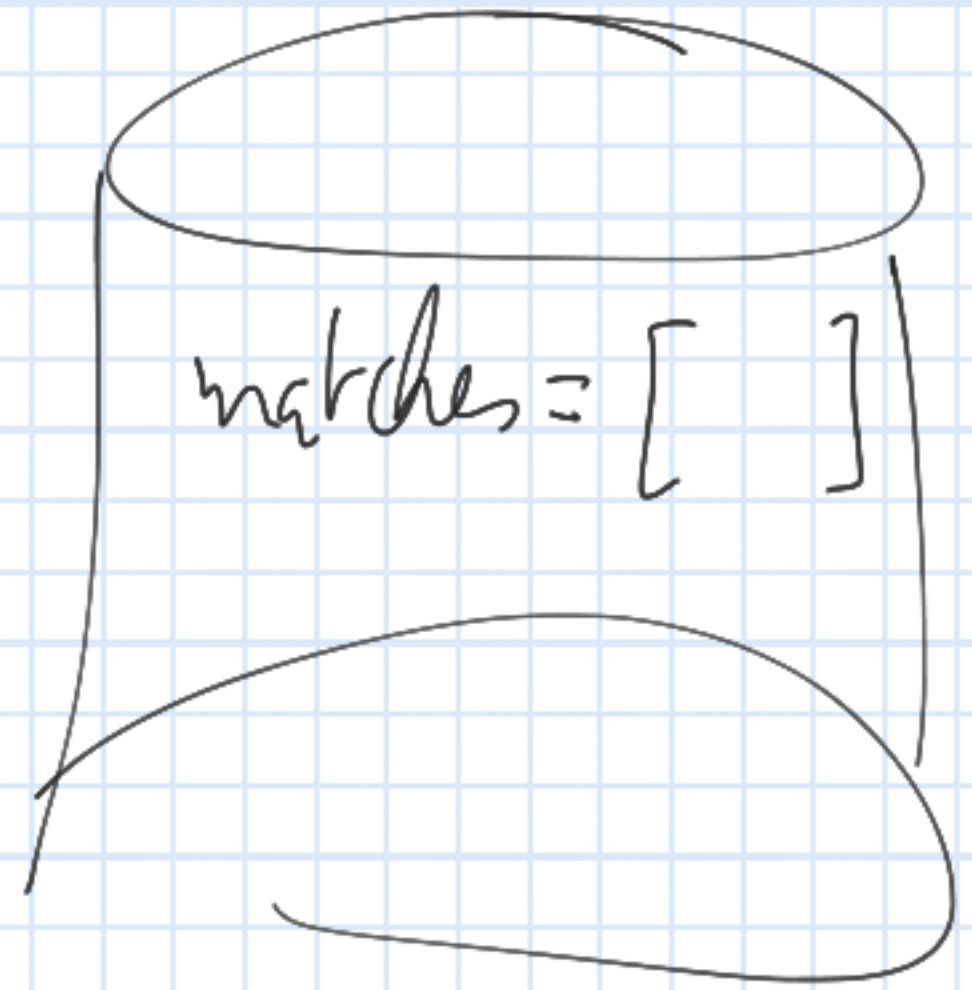
matches.push(x);

res.json({ msg: "OK", isAdded: true })

);

BodyParser

doivent être déclarées  
ds le Service



```
httpClient.post(<{msg: string, isAdded: bool}>
```

```
("http://localhost:3000/matches", obj);
```

App Service ds un Component

•) Déclarer / injecter le service ds le

•) App constructor

•) de la méthode du service

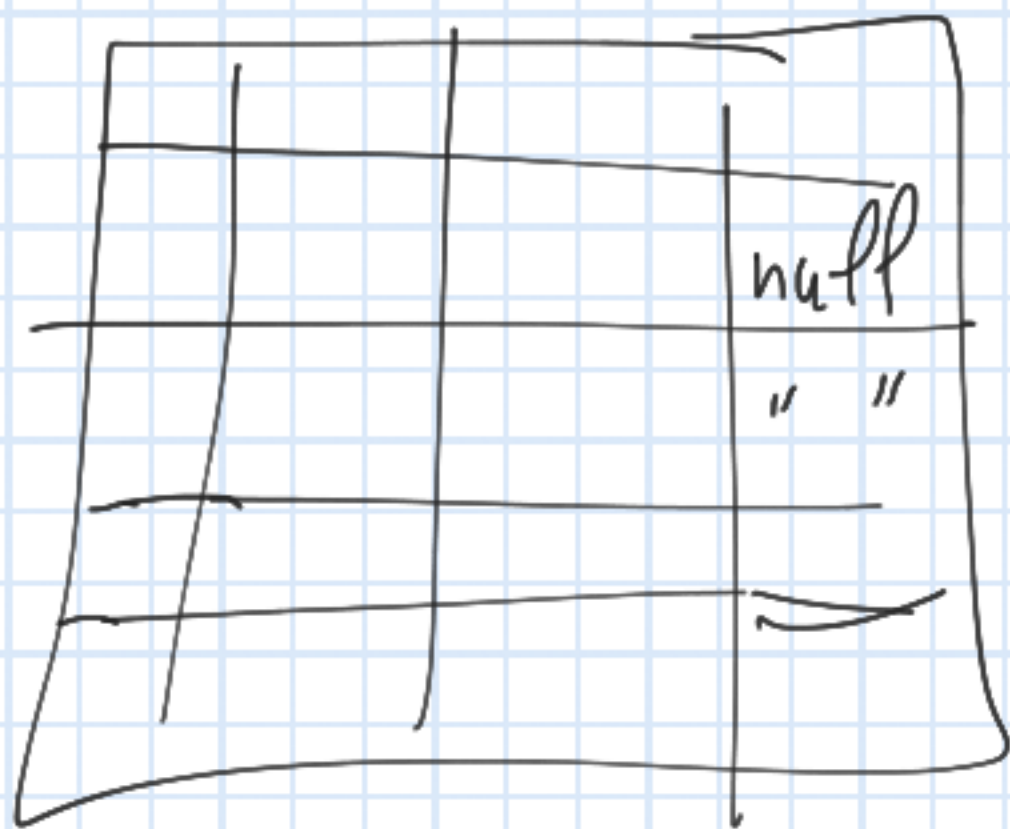
•) subscribe ( ) → récupérer la réponse



DB : espace de stockage de données

SQL

NoSQL



			null
			" "

ensemble  
de tables  
↓  
ensemble  
de lignes  
et de colonnes

ensemble de  
collections

↓  
ensemble de  
documents



Compo — r : ?

Service

getMatchBy sd (      ) {

}



Displacement

under-navigate [ ~ tech info / \$ bid ]

# Match Info

Recupera l'id

Activated Route

→ Service

getMatchBySD( x ) {

httpClient.get('http://L:2001/hd/\$x')

# Fluid

 ~~$\{h_1, h_2, h_3, h_4, h_5\}$~~

app.get("/matches/myId", (req, res) => {

// let id = req.params.myId

});

FE

Act Route.  
swap the  
pc Map.  
get('..')



Edit Match G — t

1) Remplit le formulaire  
( par OnUnit(1) )

Get id f Path ( checked Rank )

[ 1, 1, 1, 1, 1, 1 ]

for

~~id~~



app of ut ('/matches',  
(ref, res) => {  
log (ref.body)  
↓  
new values