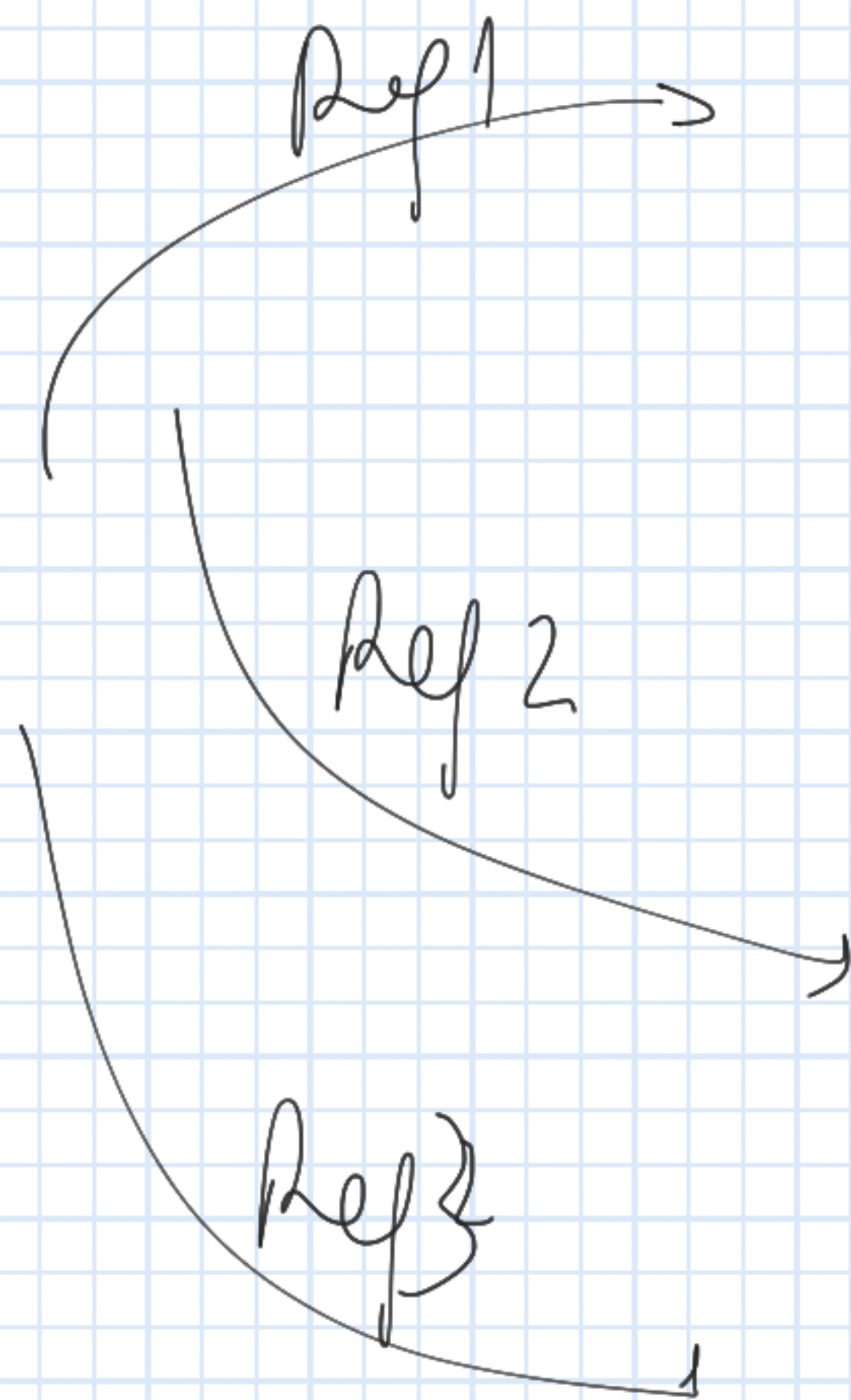


\$ ng g s players ✓  
user ✓  
match ✓  
team

Put (@, obj)  
Get (@)  
Post (@, obj)  
Delete (@)

httpClient  
↓  
Method/Action  
→ @Destination



Server.js

- listen (3000)

http://: 3000

app.js  
Business logic



app.js: Enable do f-les

app.Method HTTP ( path , (req, res) => {  
    post /matches

//

});

service

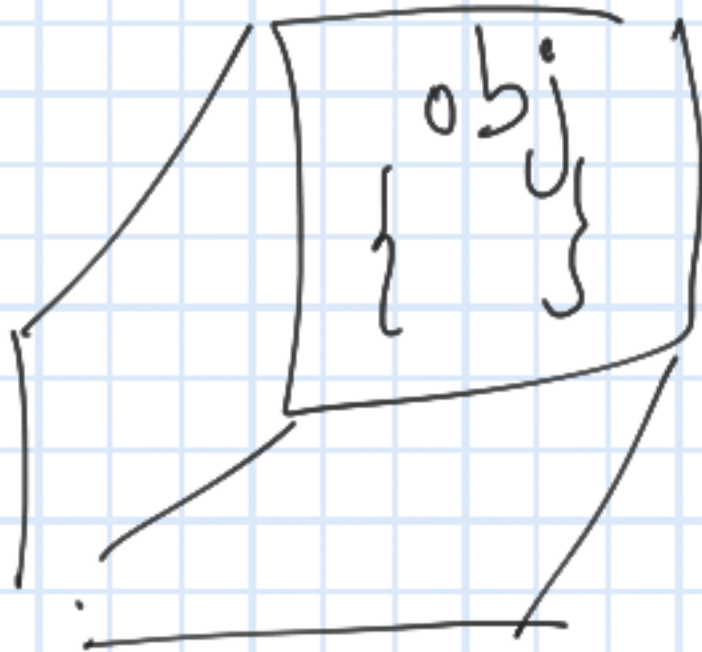
httpClient.post (@ /matches,

obj)

Post (@serverBE, obj) Request

put (@serverBE, obj)

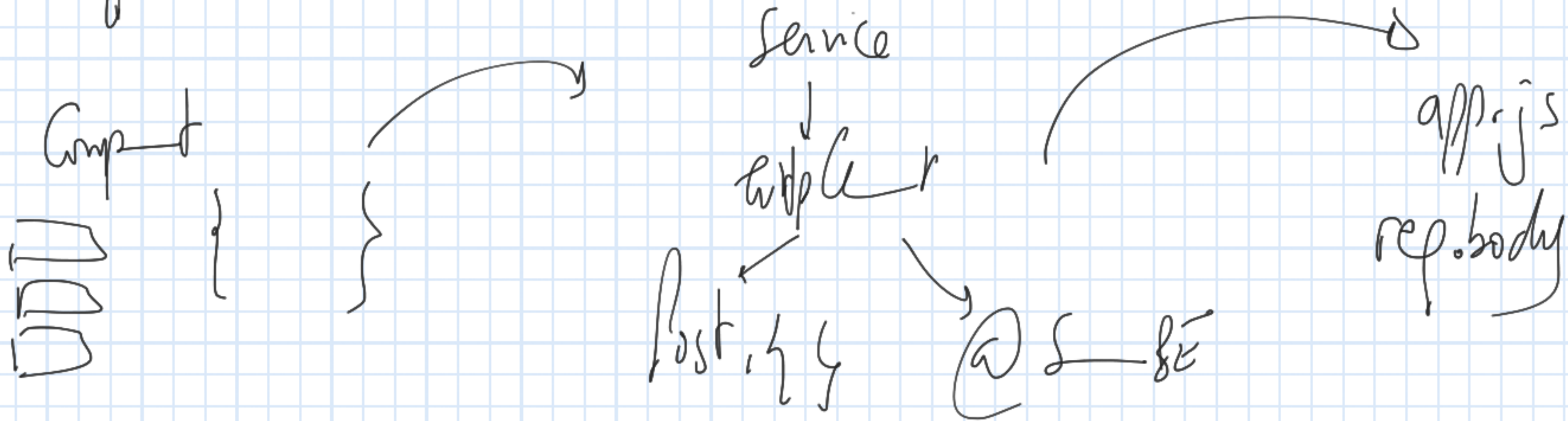
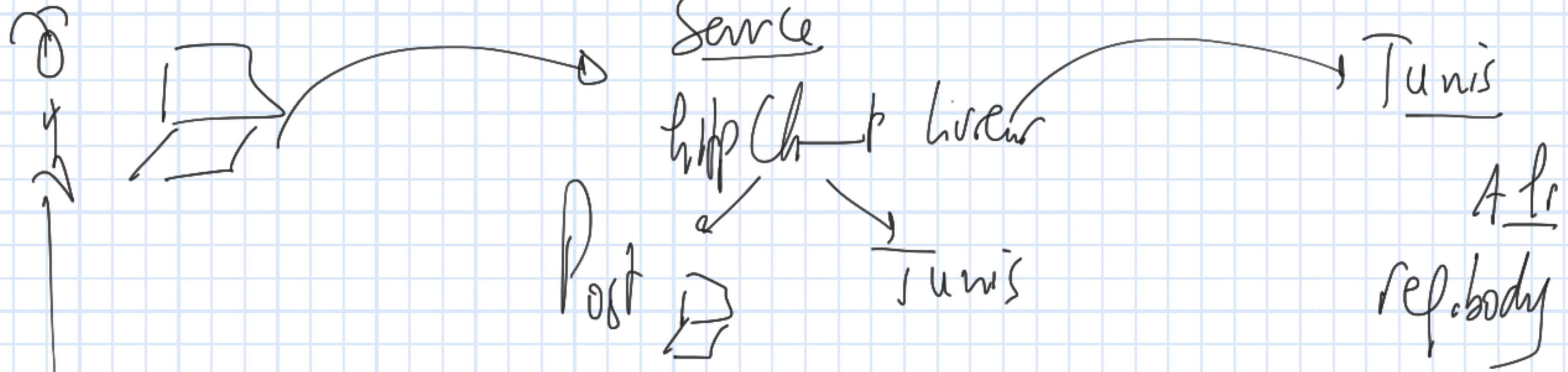
httpClient



app.js

req.body

a





Injecter un Service ds un Component :

1) déclarer ds le constructor

2) Appl de la méthode

3) `subscribe()`

↓  
Récupérer la réponse du service.

let matchesTab = [ { s1: { s1: { s1: { s1: { s1: {

let match = {

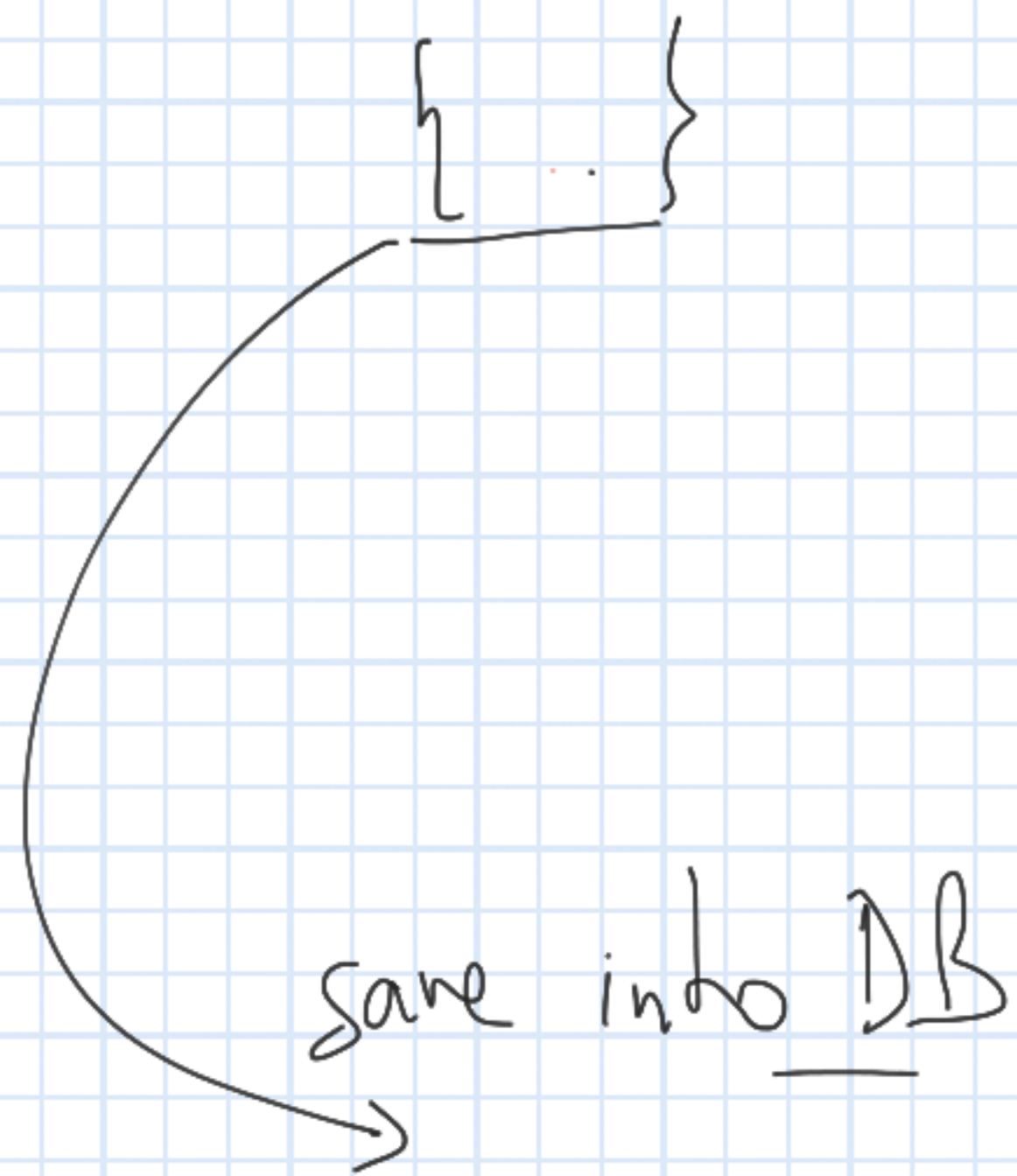

save into DB

Espace de stockage  
des données

Familles

SQL

NoSQL

Res.json({message:  
"added"})

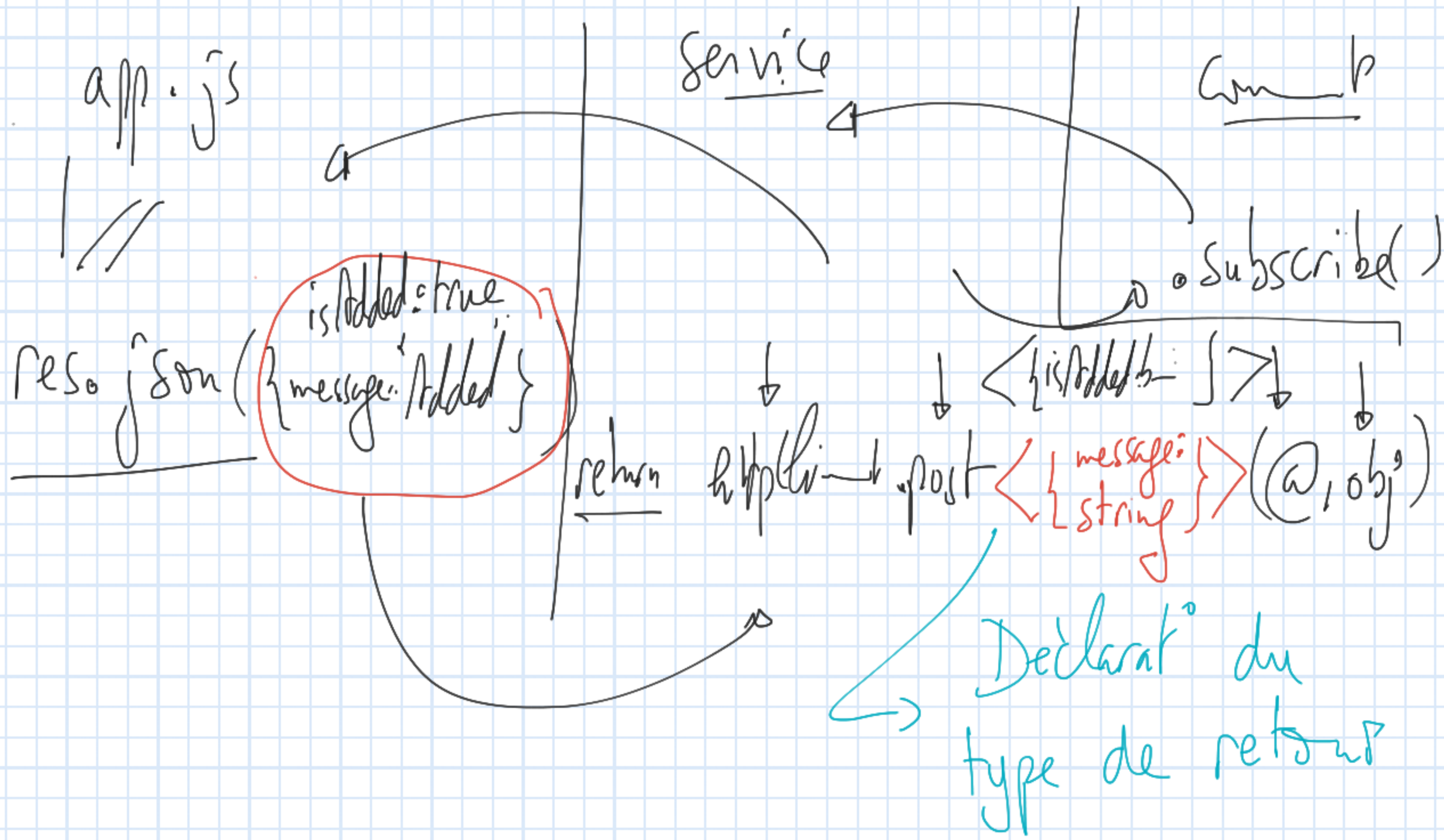
↑  
BodyParser

Res

matchesTab=[  
matchesTab.push(h)  
];







Subscribe ( $\overset{x}{\text{response data}} \Rightarrow \{ \log(\text{response}) \}$ )

pr récupérer  
la réponse  
du service

↑  
paramètre

function  
Arrow  
fléchée  
Function

message  
is added

function N\_f (params) {

// Trait

}

(A)

function (params) {

// Trait

}

func

Anonymous (B)

s'exécute 1 seule fois

||

(params) => {

}

Arrow (C)



// Get All Matches:

matches

| matches-table

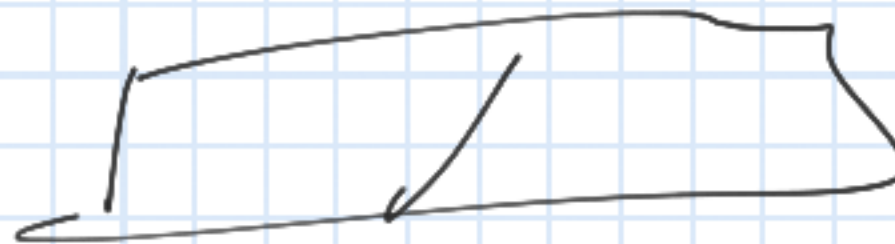
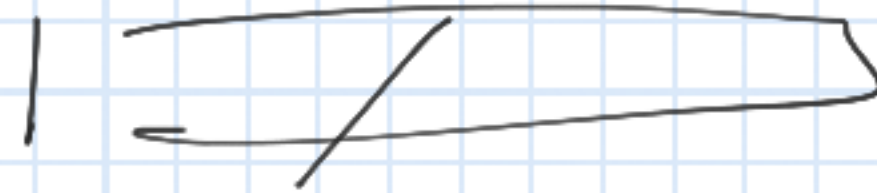
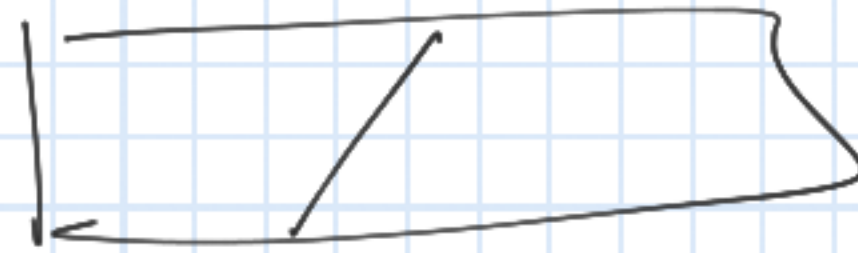
~ /all matches

Matches Count

[2, 5, 11, 15]

app-score

@input()





Test

Générer un cont nommé search

