

## Express API - TP 04 - Pokedex

### 01 . Un jour je serais le meilleur dresseur !

L'objectif sera de gérer la partie Dresseur de notre API. Chaque utilisateur ne sera associé qu'à un seul dresseur. Dans un soucis d'éviter de modifier l'existant, nous partiront de l'idée que le `Trainer` contiendra le nom de l'utilisateur pour faire le lien (et vérifier s'il y a déjà une liaison ou non).

Le modèle `Trainer` sera donc implémenté avec plusieurs attributs :

Trainer
<ul style="list-style-type: none"><li>-ObjectId id</li><li>-String username;</li><li>-String imgUrl;</li><li>-String trainerName;</li><li>-Date creationDate;</li><li>-List pkmnSeen</li><li>-List pkmnCatch;</li></ul>

Les types sont données comme indications. Vous pouvez adapter à votre convenance.

La seconde étape sera de réaliser le CRUD (Create Read Update Delete) de notre modèle. Utilisez la route `/trainer` avec les différents verbes HTTP (POST,GET,PUT,DELETE) et celles-ci demanderont forcément d'être authentifié. Vu que l'on se basera sur l'utilisateur pour savoir quel `Trainer` gérer, pas besoin de regarder les permissions

Petite aide pour récupérer les infos de l'utilisateur, pensez à votre middleware d'authentification

## 02 . Car il faut tous les attraper

Pour terminer, créez une route qui va permettre d'ajouter un Pokémon (il faut vérifier qu'il est valide) à notre dresseur. A l'aide d'un paramètre, `isCaptured` par exemple, il vous permettra de savoir dans laquelle des 2 listes il faut l'ajouter (Vu ou capturé).

Voici la route à utiliser : `/trainer/mark`

Nous avons maintenant la base nécessaire pour notre API. Vous pourrez évidemment ajouter des suppléments qui vous paraîtront utiles pour votre appli front.

Bien joué !

## 03 . Sacha et Pikachu : C'est fini

Nous en avons fini avec les fonctionnalités principales. Si lors de la réalisation de votre front, vous estimez avoir besoin d'autres routes, n'hésitez pas.

CEPENDANT (à lire avec la voix de dumbledore)

Il manque une documentation à notre API. Les gens qui vont vouloir l'utiliser n'iront pas directement regarder le code. (Bon ici, c'est toujours vous ^^)

Implémentez donc un OpenApi/Swagger pour la documentation générale. Puis, si le temps vous le permet, vérifiez bien que vos fonctions sont documenté au format [JSDoc](#)