LocateMe - Part 3 / Consignes

L'objectif de cette dernière partie de LocateMe est d'enregistrer tous les marqueurs en base de données afin de les récupérer lorsque l'application est relancée.

Tu vas être confronté à une mécanique essentielle dans le développement d'applications mobiles : la communication entre le frontend et le backend.

🖥️ Côté Backend

👉 Crées un dossier backend.

👉Dans ce dossier backend génères un setup express de base avec la commande :

express —no-view —git ./

👉 Crées les 3 routes décrites ci-dessous en suivant les exemples de requêtes et de réponses. Tu devras intégrer Mongoose à ton backend et modéliser une collection chargée d'enregistrer tous les marqueurs d'un utilisateur en base de données.

- POST /places : ajout d'un marqueur en BDD (via req.body).

Exemple de requête : POST /places

```
{ nickname: 'Max', name: 'Lyon', latitude: 45.758, longitude: 4.835 }
```

Exemple de réponse :

```
{ result: true }
```

- GET /places/:nickname : récupération de tous les marques d'un utilisateur en fonction de son surnom (via req.params).

Exemple de requête : GET /places/Max

Exemple de réponse :

```
{ result: true, places: [{ nickname: 'John', name: 'Lyon', latitude: 45.758, longitude: 4.835 }, ...] }
```

- DELETE /places : suppression d'un marqueur à partir de son nom et du surnom de l'utilisateur (via req.body).

Exemple de requête : DELETE /places

```
{ nickname: 'Max', name: 'Lyon' }
```

Exemple de réponse :

```
{ result: true }
```

Préparation des requêtes vers le backend

Avant de te détailler les interactions avec le backend pour chaque écran, tu dois être au fait d'une information importante vis-à-vis des fetchs dans une application React Native.

En effet, l'application étant démarrée sur un appareil mobile, il n'est plus possible de requêter le backend via l'URL http://localhost:3000. Il sera nécessaire de remplacer "localhost" par l'adresse IP locale de l'ordinateur dans lequel se trouve le backend.

L'adresse IP locale de ton ordinateur peut être récupérée lors du lancement de l'application via Expo, en dessous du QR code.

Il ne te resteras plus qu'à requêter le backend avec cette adresse IP locale (à la place de localhost), comme montré dans l'exemple ci-dessous.

```
fetch('http://192.168.10.110:3000/places/John')
 .then((response) => response.json())
 .then((data) => {
  data.result && dispatch(importPlaces(data.places));
 });
```

👉 Modifies le fonctionnement de MapScreen afin que lorsqu'un nouveau marqueur est ajouté (lors d'un appui long sur la map), la route POST /places soit appelée afin de l'enregistrer en base de données.

👉 Toujours sur MapScreen, fais en sorte que les marqueurs soient récupérés depuis la route GET /places/:nickname et affichés sur la map.

💡 Un peu d'aide ?

👉 Modifies le fonctionnement de PlacesScreen afin que lorsqu'une nouvelle ville est ajoutée (depuis l'input reliée à l'API Adresses), la route POST /places soit appelée afin de l'enregistrer en base de données.

👉 Toujours sur PlacesScreen, fais en sorte que les marqueurs puissent être supprimés via la route DELETE /places en plus d'être retirés du store.

Bonne chance 👨‍💻