

Deuxième partie:

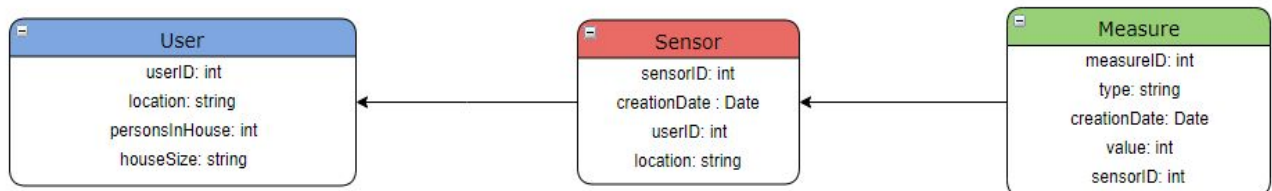
La deuxième partie concerne le back-end de votre projet.

Mise en situation

Félicitations vous avez créé un objet connecté destiné au grand public, vous avez quelques exemplaires en circulation et ils remontent déjà de la donnée ! Ces objets permettent entre autre de capturer de la donnée sur l'environnement qui les entourent. Il peuvent relever la température, l'humidité et la pollution de l'air.

Lors de notre dernière rencontre nous avons réfléchi tous ensemble à une structure de donnée pour pouvoir accéder et modifier la donnée de la meilleure des manières. Après avoir synthétisé tous nos échanges, je vous transmets la première version de cette structure de données. Elle vous aidera sûrement à travailler sur la première version de votre API qui vous permettra de consommer cette donnée en toute sécurité.

Base de donnée:



User:

- userID: Identifiant de l'utilisateur
- Country: pays de l'utilisateur
- personsInHouse: nombre de personnes dans la maison de l'utilisateur
- houseSize: taille de la maison (small, medium, big)

Sensor:

- sensorID: Identifiant du capteur
- creationDate: date de mise en place du capteur
- userID: identifiant de l'utilisateur à qui appartient le capteur
- location: endroit où est situé le capteur (kitchen, livingRoom, bedroom, bathroom, entrance)

Measure:

- measureID : identifiant de la mesure
- Type: type de mesure (humidity, temperature, airPollution)
- creationDate : date de creation de la mesure
- Value: valeur chiffrée de la mesure
- sensorID: identifiant du sensor à qui appartient cette mesure

API

Pour chacun des éléments de la base (User, Sensor, Measure) l'utilisateur de votre API doit être capable d'effectuer au minimum les quatre opérations CRUD (création, recherche, suppression, mise à jour). Sur ces éléments de la base vous devrez effectuer d'autres opérations qui vont permettront de récupérer des éléments statistiques par exemple :

- Le nombre total de capteurs en circulation
- Le nombre total d'utilisateurs
- La moyenne de température sur tout le parc des capteurs
- Les maximales/minimales des 3 derniers mois par pays
- ...

Cette liste est non exhaustive, à vous de trouver des données statistiques intéressantes à afficher dans votre dashboard. Ces éléments statistiques seront consommés par votre application de dashboard et vous aurez à les afficher dans les composants que vous avez créé dans la première partie. Il est bien sur possible de créer de nouveaux widgets afin d'avoir une analyse plus percutante des données.

Administration

Dans votre front end, créer une route `"/admin"` (à l'aide de react-router, comme vu en cours). Dans cette vue, vous aurez un formulaire qui permet de rajouter un utilisateur dans votre base et une liste de tous les utilisateurs présents dans votre base.

Objectifs obligatoires partie 2:

- ☐ Créer une API à l'aide de express qui permettra de consommer les données
- ☐ Créer une route dans votre frontend qui permettra une administration simple des utilisateurs.

Technos:

Back: Node.js, Express, mongodb, mongoose

Bonus

Vous pouvez si vous souhaitez aller plus loin, mettre en place l'une ou toutes ces fonctionnalités:

- Pouvoir ajouter des capteurs à un utilisateur
- Pouvoir modifier un utilisateur déjà existant
- Pouvoir supprimer un utilisateur (supprimer un utilisateur implique de supprimer aussi ses capteurs et les mesures de ces capteurs)