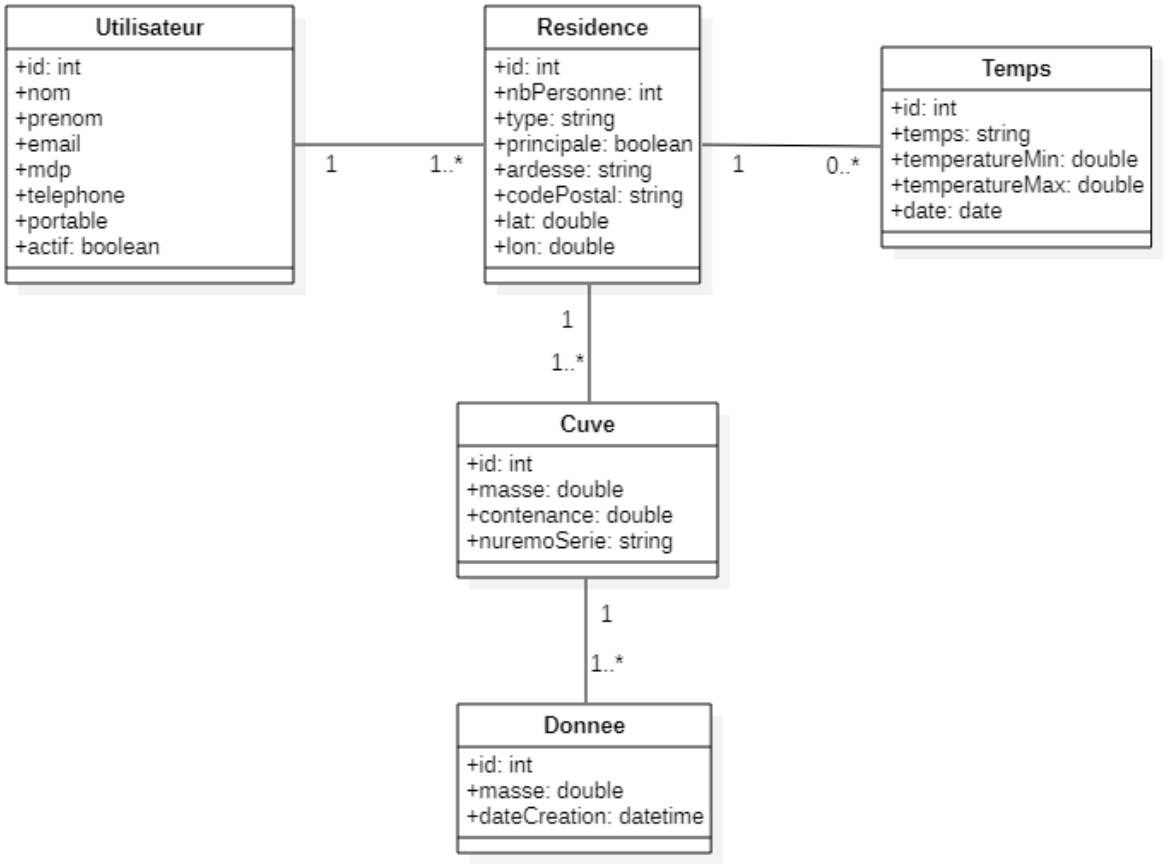
TP API Récupérateur d'eau

Énoncé

Vous allez intervenir dans le cadre d'un projet permettant de suivre la quantité d'eau disponible dans des récupérateurs d'eau. Ce projet est composé d'une application Web, d'une application Mobile, et d'un module connecté à la cuve, permettant de connaitre sa masse afin de connaitre la quantité d'eau restante.

Voici le diagramme de classe des modèles est le suivant :



Travail demander

1. Créer un nouveau projet Spring boot

2. Paramétrer le projet pour qu'il puisse utiliser une base de données MySQL ou MarianDB

3. Créer les modèles

Pour matérialiser les relations vous allez devoir rajouter dans les modèles des attributs avec les annotations @OneToMayn et @ManyToOne. Exemple avec les modèles Utilisateur et Residence :

```
@OneToMany(mappedBy="utilisateur", fetch = FetchType.LAZY)
@JsonManagedReference
private List<Residence> residences;
```

Inversement le model Residence indiquera la relation avec Utilisateur :

```
@ManyToOne(fetch = FetchType.LAZY)
@JsonBackReference
private Utilisateur utilisateur;
```

Les annotations @JsonManagedReference et @JsonBackReference sont indispensable pour éviter lorsque l'on a des relations bidirectionnelles, la récursivité infinie des données JSON.

4. Créer les repositories

5. Créer le Model ErrorValidation

6. Créer la classe RestApplicationExceptionHandler qui permettra de mettre en place notre système de validation personnalisé.

7. Créer les controllers pour les API permettant à minima de pouvoir (Tester chacune de vos API avec Postman et faire une capture d'écran pour chaque test) :

Pour les utilisateurs :

- Récupérer la liste des utilisateurs.
- Récupérer un utilisateur suivant son id
- Récupérer un utilisateur suivant sont adresse mail et son mot de passe.
- Créer un utilisateur
- Mettre à jour un utilisateur
- Supprimer un utilisateur

Pour les résidences :

- Récupérer la liste des résidences d'un utilisateur (vous aurez besoin de rajouter la méthode findByUtilisateur(Utilisateur utilisateur) dans le repository Residence, qui renvoie la liste de résidence suivant un utilisateur).
- Récupérer une résidence suivant son id
- Créer une résidence

- Mettre à jour une résidence
- Supprimer une résidence

Pour les cuves :

- Récupérer la liste des cuves d'une résidence (prévoir une méthode dans le repository Cuve).
- Récupérer une cuve suivant son id
- Créer une cuve
- Mettre à jour une cuve
- Supprimer une cuve

Pour les données :

- Récupérer la liste des données d'une cuve (prévoir une méthode dans le repository Donnee).
- Récupérer une donnée suivant son id
- Créer une donnée
- Mettre à jour une donnée
- Supprimer une donnée

Pour les temps

- Récupérer la liste des 5 derniers jours de temps d'une résidence (prévoir une méthode dans le repository Temps).
- Créer un temps
- Mettre à jour un temps

Bonus :

8. Pour le connaitre le temps vous aller utiliser l'API OpenWeather https://openweathermap.org/api/one-call-3 qui permet de connaitre le temps qu'il fait n'importe où dans le monde. L'API key pour utiliser cette API est la suivante : fe15b043f8750eff8eca663f0b86a299

   On utilisera la fonctionnalité https://openweathermap.org/api/geocoding-api qui permettra de récupérer la latitude et la longitude d'une adresse en indiquant le nom de la ville et le pays. Exemple : https://api.openweathermap.org/geo/1.0/direct?q=lyon,fr&limit=1&appid=fe15b043f8750eff8eca663f0b86a299

   Et la fonctionnalité https://openweathermap.org/forecast5 qui permettra de connaitre le temps sur 5 jours. Exemple : https://api.openweathermap.org/data/2.5/forecast?lat=45.7578137&lon=4.8320114&lang=fr&appid=fe15b043f8750eff8eca663f0b86a299

   Pour appeler une API dans un projet Spring vous aller devoir utiliser la classe RestTemplate : https://www.baeldung.com/rest-template https://reflectoring.io/spring-resttemplate/

   Créer un Service WeatherApiService qui contiendra 2 méthodes :

- Une méthode qui appellera l'api pour récupérer la latitude et la longitude d'une résidence et qui mettra à jour les 2 champs de cette résidence dans la base de données.
- Une méthode qui appellera l'api pour récupérer le temps qu'il fera sur 5 jours pour une résidence passée en paramètre et qui ajoutera ces informations dans la table Temps