

Note Explicative sur les Choix Techniques

Choix de l'ORM/ODM : Mongoose

J'ai choisi d'utiliser Mongoose pour plusieurs raisons :

- **Utilisation du NoSQL** : Je n'ai pas utilisé de Base de données relationnelle, et, étant donné que nous avons vu en cours Mongoose, j'ai préféré cette alternative. Evidemment, le projet semble plus correspondre à une base de données relationnelle (avec MySQL et TypeORM notamment), mais j'ai voulu assurer cette partie-là étant tout seul pour le projet.
- **Modélisation Simplifiée** : Schémas stricts pour une structure de données cohérente.
- **Validation Intégrée** : Réduit les erreurs grâce à des validations robustes.

Gestion des Erreurs

1. Centralisation des Erreurs

- Utilisation de ErrorResponse et ErrorResponse pour structurer les erreurs avec :
 - Code HTTP
 - Code d'erreur dédié
 - Message clair.

2. Validation avec Class-Validator

- Validation des données entrantes (DTOs) avant leur persistance.

Structure du Projet

Organisation Modulaire

- **Controllers** : Gèrent les routes et orchestrent les services.
- **Services** : Contiennent la logique métier.
- **Repositories** : Accès direct à la base via Mongoose.
- **Utils** : Outils génériques (gestion des erreurs, réponses, etc.).

- **DTOs, Inputs et Presenters :**

- DTOs : Structurent les données entrantes.
- Presenters : Structurent les données sortantes.

Exemple : Création d'une Colocation

1. **Controller** : Valide les données et appelle le service.
2. **Service** : Applique la logique métier.
3. **Repository** : Interagit avec MongoDB via Mongoose.
4. **Response** : Retourne les données formatées au client.