

Systeme de Gestion d'Hôtel

Ce projet a pour objectif de développer une application API web en C# pour la gestion d'un hôtel, couvrant divers aspects comme la réservation de chambres, la gestion des clients, le personnel et les services de l'hôtel.

L'objectif est d'appliquer les concepts de programmation orientée objet, les patterns d'architecture et de conception, et de développer une application fonctionnelle et sécurisée.

Execution du projet

Mettre le projet HotelManagement.Api.UI en projet de démarrage et le lancer avec le https pour que le swagger s'ouvre sous l'url <https://localhost:7080/swagger/index.html>

une fois cela fait , il faut importer la collection postman presente dans le projet HotelManagement.Api.UI (hotel management.postman_collection.json)

dans ce json il y a les routes d'auth avec 3 requetes

ce connecter avec le compte client

ce connecter avec le compte receptionniste

ce connecter avec le compte personnel de ménage

Ensuite les requete lié au chambre et les requete lié au reservations et au payment

Architecture

Une Clean archi DDD a été faite pour ce projet avec différents patterns :

- factory pattern
- strategy pattern
- repository pattern
- singleton pattern
- un système de useCase mis en place qui fait office de command pattern (pseudo CQRS)
- respect des principes SOLID

Stockage

Pour ce projet un système de persistance de données en json a été choisi pour éviter de prendre trop de temps sur la connexion bdd avec EF core

On a simulé un contexte comme EF core qui nous permet de faire un CRUD sur nos données

Amélioration

Mise en place d'une bdd Sqlserver ou postgresSQL et installation et configuration de EFcore pour mapper nos modèles

avec nos tables en BDD

Mise en place d'un Unit of work

amélioration du repository pattern

Utilisateur

il y a une route de création des utilisateurs pour tester nos routes avec un rôle associé à chaque utilisateur

Le mot de passe est crypté grâce à la méthode PasswordHasher fournie par dotnet

Un système de Login est fait qui renvoie un bearer Token et les endpoints sont soumis à un système d'autorisation et

même parfois d'accessibilité selon le rôle de l'utilisateur

Défis Rencontrer

- Apprentissage du c#
- Apprentissage dotnet
- Charge de travail et timing

Axe amelioration

- Mise en place du vrai system de commande pattern
- Mise en place du CQRS avec MediaTor
- Mise en place TDD
- Envoie de notifications
- Creer une interface front end