Compte rendu Hôtelier

Voici mon schéma UML de niveau 1 :

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

Voici mon schéma UML de niveau 2 :

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

3 )

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement Voici mes trois collections

Pour ajouter des données dans mes collections j’ai en premier lieu crée 3 classes(Client, Chambre, Reservataire).

Maintenant je vais instancié deux clients, deux Chambre, et deux Reservataire dans mon Main :

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquementJe vais me connecter à ma base :

Une image contenant texte, ligne, capture d’écran, Police

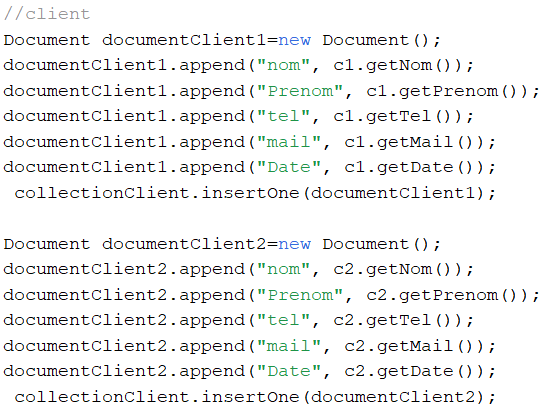
Description générée automatiquement

Après je vais charger ma base, nommé Hotelinter et mes collections :

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

Pour ensuite instancier mes clients, mes chambres et mes réservataires :



Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, nombre

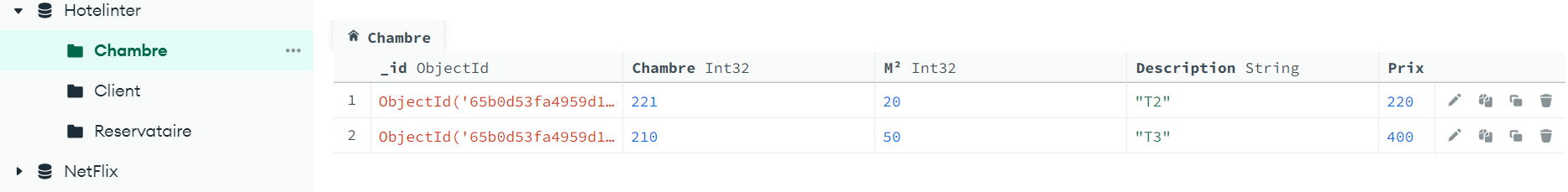
Description générée automatiquement

Une image contenant texte, capture d’écran, Police, ligne

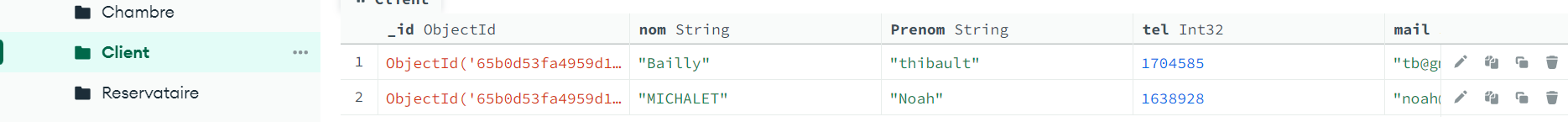
Description générée automatiquement

Je vais vérifier dans ma base pour les résultat :

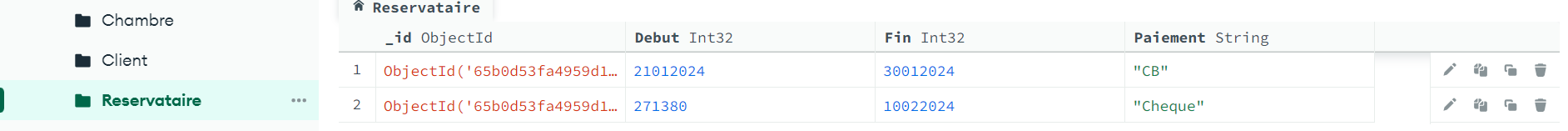
Ma collection chambre :



Ma collection Client :



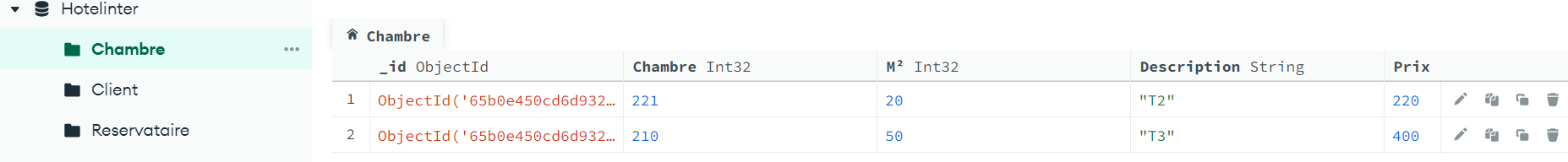
Ma collection Reservataire :



Pour supprimer les clients j’utilise la commande



Quand je vais lancer mon programme ma base MongoDB va passer en Drop and Create en supprimant les anciennes données pour crée les nouvelles.

Alors que dans ma collection Reservataire je possède encore les ancienne données

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Après la création des collections je vais maintenant crée les associations

Pour cela je commence par crée un nouvelle attribut nommé ‘id’ de type ObjectID, dans mes trois classes.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquementExemple :

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

Je répète l’opération dans ma collection Chambre et Réservataire.

Seconde étape, je me rends dans mon starter et je crée mes listes

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La création de liste va permettre de stocker plusieurs Chambres, Clients et réservataires.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Ensuite j’ai utilisé les quatre premières lignes pour ajouter une réservation dans *client1*, *client2*, *chambre1* et *chambre2*.

Les deux dernières lignes servent à ajouter *Chambre1* dans Reservataire1 et *chambre2* dans *Reservataire2*.

Troisième étape, je vais ajouter Réservataire dans mon *documentClient1*

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Je répète l’opération pour *Client2.*

Pour le reste des collections j’ai procéder de manières différente :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

J’ajoute *Reservataire* dans ma *Chambre1* et ma création de liste va permettre de parcourir l’ID dans la collection *Reservataire.*

Je répètes l’opération pour le reste de mes collections.

Quatrième étape, créez des liens entre les objets Java et les enregistrements MongoDB.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

Exemple avec *Client1* et *Reservataire1* :

***c1.setId((ObjectId) documentClient1.get("\_id"));*** : J’associe l'objet *c1* avec le document MongoDB correspondant. Je récupère l'ID du document *documentClient1* dans la base de données MongoDB et l'attribuez à l'attribut ID de l'objet *c1*.

**r1.setObj((ObjectId) documentReservataire1.get("\_id"));** : J’associe l'objet *r1* avec le document MongoDB correspondant. Je récupère l'ID du document *documentReservataire1* dans la base de données MongoDB et l'attribuez à l'attribut obj de l'objet *r1*.

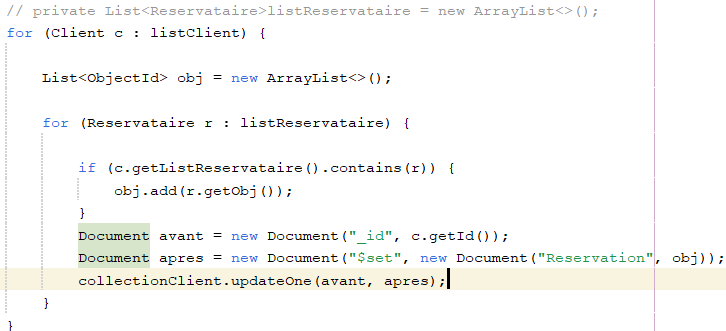
J’obtiens un résultat comme l’image ci-dessous :

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Dernière étape, je vais donc insérer les OID a la place des guillemets

Pour cela j’utilise une boucle *for* quimet à jour la collection des clients dans MongoDB en associant les réservations appropriées à chaque client, en remplaçant la liste existante de réservations par une nouvelle liste d'ObjectId.



Voici le resultat sur MongoDB:

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Collection Client



Collection Reservataire

Comme on peut le constater mon *Cient1* et associé à *Réservataire1* car Client1 possède l’ID de Reservataire.

Je répètes les mêmes opérations que j’ai effectué pour Client et Reservataire et j’obtiens un resultat final comme ceci :



Dans ma collection Reservataire j’ai bien un client associé ainsi que la chambre.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Dans ma collection Chambre j’ai bien la reservation associé

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Dans ma collection Client j’ai la reservation associé.