



MASTER
INFORMATIQUE

Conversion du MCD MERISE en des objets Couchbase

Membres

Fayssal EL ANSARI
Jonathan DI MARTINO

15/06/2023

1. Spécification des modèles de documents à mettre dans chaque collection

Appartement : Cette entité serait un document avec les attributs suivants : `_id`, `price`, `location`, `numberOfRooms`, `numberOfBathrooms`, `_class`.

Exemple de document en JSON :

```
json Copy code

{
  "_id": "app1",
  "_class": "Appartement",
  "price": 500,
  "location": "Paris",
  "numberOfRooms": 3,
  "numberOfBathrooms": 2
}
```

Clients : Cette entité serait un document avec les attributs suivants : `_id`, `email`, `password`, `phoneNumber`, `countryOfOrigin`, `isVerified`, `_class`.

Exemple de document en JSON :

```
json Copy code

{
  "_id": "client1",
  "_class": "Client",
  "email": "john.doe@example.com",
  "password": "securePassword",
  "phoneNumber": "0123456789",
  "countryOfOrigin": "France",
  "isVerified": true
}
```

Reservation : Cette entité serait un document avec les attributs suivants : `_id`, `apartmentId`, `userId`, `startDate`, `endDate`, `price`, `_class`.

Exemple de document en JSON :

```
json Copy code

{
  "_id": "res1",
  "_class": "Reservation",
  "apartmentId": "app1",
  "userId": "client1",
  "startDate": "2023-06-15",
  "endDate": "2023-06-20",
  "price": 500
}
```

Transaction : Cette entité serait un document avec les attributs suivants : `_id`, `clientId`, `reservationId`, `amount`, `transactionType`, `transactionStatus`, `_class`.

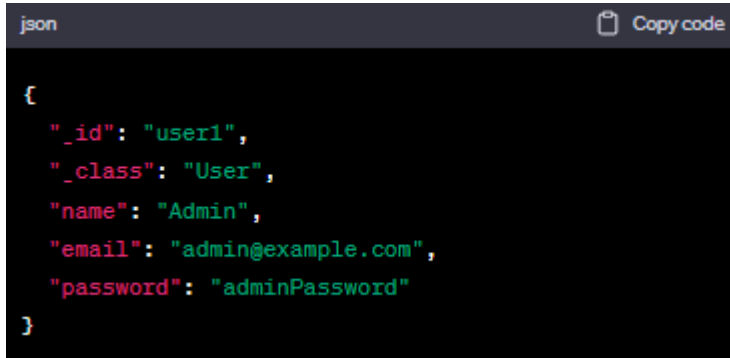
Exemple de document en JSON :

```
json Copy code

{
  "_id": "trans1",
  "_class": "Transaction",
  "clientId": "client1",
  "reservationId": "res1",
  "amount": 500,
  "transactionType": "debit",
  "transactionStatus": "completed"
}
```

Utilisateur : Cette entité serait un document avec les attributs suivants : `_id`, `name`, `email`, `password`, `_class`.

Exemple de document en JSON :



```
json Copy code

{
  "_id": "user1",
  "_class": "User",
  "name": "Admin",
  "email": "admin@example.com",
  "password": "adminPassword"
}
```

2. Spécification des classes et des méthodes JAVA

Chaque classe Java correspondante aurait les méthodes CRUD suivantes :

`create()` : pour insérer un nouveau document dans la base de données.

`creates()` : pour insérer un ou plusieurs nouveau documents dans la base de données.

`get()` : pour lire un document à partir de la base de données en utilisant l'ID du document.

`getAll()` : pour lire un ou plusieurs documents à partir de la base de données en utilisant l'ID du document.

`update()` : pour modifier un document existant dans la base de données.

`updates()` : pour modifier un ou plusieurs documents existant dans la base de données.

`delete()` : pour supprimer un document de la base de données.

`deletes()` : pour supprimer un ou plusieurs documents de la base de données.

Pour les indexes secondaires, avec Couchbase il faut un index primaire sur la collection pour exécuter des requêtes N1QL sur une collection. Voici les requêtes pour créer un index primaire sur chaque Collection

```
CREATE PRIMARY INDEX ON `mtest`.`tester`.`Apartments`;  
CREATE PRIMARY INDEX ON `mtest`.`tester`.`Clients`;  
CREATE PRIMARY INDEX ON `mtest`.`tester`.`Reservation`;  
CREATE PRIMARY INDEX ON `mtest`.`tester`.`Transactions`;  
CREATE PRIMARY INDEX ON `mtest`.`tester`.`Users`;
```

En ce qui concerne les méthodes applicatives de consultation, on va utiliser le langage de requête N1QL et l'API Java pour Couchbase. Donc les méthodes vont être répartis entre les opérations sans N1QL et avec N1QL. Voici les opérations.

Pour les opérations sans N1QL:

- ✓ Recherche d'appartements par utilisateur

```
SELECT * FROM `mtest`.`tester`.`Apartments` WHERE ownerId =  
$userId"
```

- ✓ Recherche de transactions par client

```
SELECT * FROM `mtest`.`tester`.`Transactions` WHERE clientId =  
$clientId
```

- ✓ Recherche de réservations par appartement

```
SELECT * FROM `mtest`.`tester`.`Reservation` WHERE apartmentId =  
$apartmentId
```

- ✓ Recherche d'appartements par nombre de chambres

```
SELECT * FROM `mtest`.`tester`.`Apartments` WHERE numberOfRooms =  
$numberOfRooms
```

- ✓ Recherche de clients par pays d'origine

```
SELECT * FROM `mtest`.`tester`.`Clients` WHERE countryOfOrigin =  
$country
```

- ✓ Trouver tous les appartements dans une certaine gamme de prix

```
SELECT * FROM `mtest`.`tester`.`Apartments` WHERE defaultPrice >=  
$minPrice AND defaultPrice <= $maxPrice
```

Pour les opérations utilisant N1QL:

- ✓ Trouver tous les utilisateurs qui n'ont pas encore vérifié leur compte

```
SELECT * FROM `mtest`.`tester`.`Users` WHERE isVerified = false
```

- ✓ Trouver toutes les réservations pour un appartement spécifique

```
SELECT * FROM `mtest`.`tester`.`Reservation` WHERE apartmentId = $apartmentId
```

- ✓ Trouver tous les clients d'un pays d'origine spécifique

```
SELECT * FROM `mtest`.`tester`.`Clients` WHERE countryOfOrigin = $country
```

- ✓ Trouver tous les appartements ayant un certain nombre de salles de bains

```
SELECT * FROM `mtest`.`tester`.`Apartments` WHERE numberOfBathrooms = $numberOfBathrooms
```