Gestion des avis (partie 05)

# TP05 du module 03 – Spring Data MongoDB

|  |
| --- |
| Ce TP sera un TP fil rouge du cours   * Création d’une association DBRef (stratégie 3) * Utilisation de tests unitaires pour valider |

|  |
| --- |
| **Durée estimée** |
| 30 minutes |

Contexte

En vous appuyant sur l’ensemble des concepts abordés dans le cours, mettez en place une application Spring Boot permettant la gestion des avis d’une cave à vin.

L’application de cave à vin sera une application d’API Web sécurisée à la fin des itérations.

L’application de gestion des avis sera aussi une application d’API Web :

* Elle doit permettre de gérer les avis des clients sur le vin commandé.
  + La date de l’avis
  + L’identifiant de la bouteille commandée, ainsi que de la Région et couleur
  + La quantité commandée par le client
  + Une note (entre 0 et 5), un commentaire et le pseudo du client (permettre de protéger les informations du client RGPD)
* Le but étant de faire des statistiques pour le propriétaire du Vignoble

Dans les itérations de ce module ; nous réaliserons uniquement du back de l’application. En particulier, la couche DAL et les BO pour le moment.

Cette application se fera avec Spring Boot.

Déclarer la structure des données en Anglais

Solution

# Supprimer les Avis et le Bouteille en base

* Dans Mongo Express, supprimer les données des 2 Collections

# Niveau 1 : Filtrer les Avis

A cette itération, nous allons ajouter des méthodes pour réaliser des requêtes spécifiques pour filtrer les Avis :

* Les Avis qui ont une note < à la note en paramètre
* Les Avis qui ont une note ≥ à la note en paramètre
* Remonter les Avis associé à une Bouteille

## Compléter AvisRepository

**package** fr.eni.gestionavis.dal;

**import** java.time.LocalDateTime;

**import** java.util.List;

**import** org.springframework.data.mongodb.repository.MongoRepository;

**import** org.springframework.data.repository.query.Param;

**import** fr.eni.gestionavis.bo.Avis;

**import** fr.eni.gestionavis.bo.vin.Bouteille;

**public** **interface** AvisRepository **extends** MongoRepository<Avis, String> {

// Les Avis qui ont une note < à la note en paramètre

List<Avis> findByNoteLessThan(@Param("note") **int** note);

// Les Avis qui ont une note ≥ à la note en paramètre

List<Avis> findByNoteGreaterThan(@Param("note") **int** note);

// Remonter les Avis associé à une Bouteille

List<Avis> findByBouteille(@Param("bouteille") Bouteille bouteille);

}

# Niveau 1 : Créer des tests unitaires

* Vous avez une classe de tests : DataPourRequetes.java qui vous est fournie
  + Copier la
  + Exécuter la
  + Vous avez maintenant en base :
    - 3 Bouteille
    - 9 Avis sur 3 Client différents

En vous inspirant des tests des démonstrations :

* Créer une classe de tests unitaires qui s’appuie sur le contexte de Spring
  + @Slf4j
  + @SpringBootTest
  + @TestMethodOrder(MethodOrderer.MethodName.**class**)
* Créer des tests pour chaque requête
  + Pour la note ; tester avec 3

**package** fr.eni.gestionavis.requetes;

**import** **static** org.assertj.core.api.Assertions.*assertThat*;

**import** java.time.LocalDateTime;

**import** java.util.List;

**import** org.junit.jupiter.api.\*;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.boot.test.context.SpringBootTest;

**import** fr.eni.gestionavis.bo.Avis;

**import** fr.eni.gestionavis.bo.vin.\*;

**import** fr.eni.gestionavis.dal.AvisRepository;

**import** lombok.extern.slf4j.Slf4j;

@Slf4j

@SpringBootTest

@TestMethodOrder(MethodOrderer.MethodName.**class**)

**class** TestRequetes {

@Autowired

AvisRepository avisRepository;

@Test

**void** test01\_findByNoteLessThan\_3() {

List<Avis> listeAvis = avisRepository.findByNoteLessThan(3);

*assertThat*(listeAvis).isNotNull();

*assertThat*(listeAvis).isNotEmpty();

*assertThat*(listeAvis.size()).isGreaterThan(1);

***log***.info("Le nombre d'Avis avec une note < 3 est de : " + listeAvis.size());

***log***.info(listeAvis.toString());

}

@Test

**void** test02\_findByNoteBouteilleGreaterThan\_2() {

// GreaterThan à 2 pour que ce soit bien note >= 3

List<Avis> listeAvis = avisRepository.findByNoteGreaterThan(2);

*assertThat*(listeAvis).isNotNull();

*assertThat*(listeAvis).isNotEmpty();

*assertThat*(listeAvis.size()).isGreaterThan(1);

***log***.info("Le nombre d'Avis avec une note >= 3 est de : " + listeAvis.size());

***log***.info(listeAvis.toString());

}

@Test

**void** test03\_findByBouteille() {

**final** Bouteille b = Bouteille

.*builder*()

.id(BouteilleId

.*builder*()

.idBouteille(18298)

.idRegion(3)

.idCouleur(1)

.build())

.nom("Vin ENI Edition")

.build();

List<Avis> listeAvis = avisRepository.findByBouteille(b);

*assertThat*(listeAvis).isNotNull();

*assertThat*(listeAvis).isNotEmpty();

*assertThat*(listeAvis.size()).isEqualTo(3);

***log***.info(listeAvis.toString());

}

}

* Voici le type de trace attendu :

find using query: { "note" : { "$lt" : 3}} fields: Document{{}} for class: class fr.eni.gestionavis.bo.Avis in collection: reviews

Le nombre d'Avis avec une note < 3 est de : 3

[

Avis(id=64be818557e18425fae61851, note=2, commentaire=Commentaire (2), date=2023-07-13T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=22), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61852, note=2, commentaire=Commentaire (2), date=2023-07-23T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=44), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service)),

Avis(id=64be818557e18425fae61853, note=2, commentaire=Commentaire (2), date=2023-08-02T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=88), bouteille=Bouteille(id=BouteilleId(idBouteille=1999, idRegion=2, idCouleur=3), nom=Vin ENI Ecole))]

find using query: { "note" : { "$gt" : 2}} fields: Document{{}} for class: class fr.eni.gestionavis.bo.Avis in collection: reviews

Le nombre d'Avis avec une note >= 3 est de : 6

[

Avis(id=64be818557e18425fae61854, note=3, commentaire=Commentaire (3), date=2023-07-13T15:28, client=Client(pseudo=patricketoile@email.fr, quantiteCommandee=36), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61855, note=3, commentaire=Commentaire (3), date=2023-07-23T15:28, client=Client(pseudo=patricketoile@email.fr, quantiteCommandee=108), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service)),

Avis(id=64be818557e18425fae61856, note=3, commentaire=Commentaire (3), date=2023-08-02T15:28, client=Client(pseudo=patricketoile@email.fr, quantiteCommandee=324), bouteille=Bouteille(id=BouteilleId(idBouteille=1999, idRegion=2, idCouleur=3), nom=Vin ENI Ecole)),

Avis(id=64be818557e18425fae61857, note=4, commentaire=Commentaire (4), date=2023-07-13T15:28, client=Client(pseudo=carlotentacule@email.fr, quantiteCommandee=100), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61858, note=4, commentaire=Commentaire (4), date=2023-07-23T15:28, client=Client(pseudo=carlotentacule@email.fr, quantiteCommandee=400), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service)),

Avis(id=64be818557e18425fae61859, note=4, commentaire=Commentaire (4), date=2023-08-02T15:28, client=Client(pseudo=carlotentacule@email.fr, quantiteCommandee=1600), bouteille=Bouteille(id=BouteilleId(idBouteille=1999, idRegion=2, idCouleur=3), nom=Vin ENI Ecole))]

find using query: { "bottle\_id" : { "$java" : { "$ref" : "bottles", "$id" : "Document{{bottle\_id=18298, region\_id=3, color\_id=1}}" } } } fields: Document{{}} for class: class fr.eni.gestionavis.bo.Avis in collection: reviews

[

Avis(id=64be818557e18425fae61851, note=2, commentaire=Commentaire (2), date=2023-07-13T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=22), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61854, note=3, commentaire=Commentaire (3), date=2023-07-13T15:28, client=Client(pseudo=patricketoile@email.fr, quantiteCommandee=36), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61857, note=4, commentaire=Commentaire (4), date=2023-07-13T15:28, client=Client(pseudo=carlotentacule@email.fr, quantiteCommandee=100), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition))]

* + Remarquer que c’est l’identifiant de la Bouteille qui est validée par la requête
  + Les autres champs ne sont pas vérifiés

# Niveau 2 : Filtrer les Avis

## 2 dernières requêtes à créer

* Remonter les Avis selon le pseudo d’un Client
* Remonter les Avis dont la quantité commandée dépasse celle en paramètre
* Remonter les Avis dont la date est entre les 2 dates en paramètres

## Compléter AvisRepository

**package** fr.eni.gestionavis.dal;

**import** java.time.LocalDateTime;

**import** java.util.List;

**import** org.springframework.data.mongodb.repository.MongoRepository;

**import** fr.eni.gestionavis.bo.Avis;

**import** fr.eni.gestionavis.bo.vin.Bouteille;

**public** **interface** AvisRepository **extends** MongoRepository<Avis, String> {

…

// Remonter les Avis selon le pseudo d’un Client

List<Avis> findByClientPseudo(@Param("pseudo") String pseudo);

// Remonter les Avis dont la quantité commandée dépasse celle en paramètre

List<Avis> findByClientQuantiteCommandeeGreaterThan(@Param("qte") **int** qte);

// Remonter les Avis dont la date est entre les 2 dates en paramètres

List<Avis> findByDateBetween(@Param("deb") LocalDateTime deb, @Param("fin") LocalDateTime fin);

}

## Créer les tests correspondants

* Tester pour le pseudo : bobeponge@email.fr
* Tester pour une quantité de 100
* Test avec le 01 Aout 2023

**package** fr.eni.gestionavis.requetes;

**import** **static** org.assertj.core.api.Assertions.*assertThat*;

**import** java.time.LocalDateTime;

**import** java.util.List;

**import** org.junit.jupiter.api.\*;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.boot.test.context.SpringBootTest;

**import** fr.eni.gestionavis.bo.Avis;

**import** fr.eni.gestionavis.bo.vin.\*;

**import** fr.eni.gestionavis.dal.AvisRepository;

**import** lombok.extern.slf4j.Slf4j;

@Slf4j

@SpringBootTest

@TestMethodOrder(MethodOrderer.MethodName.**class**)

**class** TestRequetes {

…

@Test

**void** test04\_findByClientPseudo\_bobeponge() {

String pseudo = "bobeponge@email.fr";

List<Avis> listeAvis = avisRepository.findByClientPseudo(pseudo);

*assertThat*(listeAvis).isNotNull();

*assertThat*(listeAvis).isNotEmpty();

*assertThat*(listeAvis.size()).isEqualTo(3);

***log***.info(listeAvis.toString());

}

@Test

**void** test05\_findByClientQuantiteCommandeeGreaterThan\_100() {

**int** qte = 100;

List<Avis> listeAvis = avisRepository.findByClientQuantiteCommandeeGreaterThan(qte);

*assertThat*(listeAvis).isNotNull();

*assertThat*(listeAvis).isNotEmpty();

*assertThat*(listeAvis.size()).isEqualTo(4);// 108 - 324 - 400 - 1600

***log***.info(listeAvis.toString());

}

@Test

**void** test06\_findByDateBetween() {

LocalDateTime debut = LocalDateTime.*of*(2023, 7, 13, 12, 28);

LocalDateTime fin = LocalDateTime.*of*(2023, 7, 31, 15, 28);

List<Avis> listeAvis = avisRepository.findByDateBetween(debut,fin);

*assertThat*(listeAvis).isNotNull();

*assertThat*(listeAvis).isNotEmpty();

*assertThat*(listeAvis.size()).isEqualTo(6);

***log***.info(listeAvis.toString());

}

}

* Voici le type de trace attendu :

find using query: { "client.login" : "bobeponge@email.fr"} fields: Document{{}} for class: class fr.eni.gestionavis.bo.Avis in collection: reviews

[

Avis(id=64be818557e18425fae61851, note=2, commentaire=Commentaire (2), date=2023-07-13T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=22), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61852, note=2, commentaire=Commentaire (2), date=2023-07-23T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=44), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service)),

Avis(id=64be818557e18425fae61853, note=2, commentaire=Commentaire (2), date=2023-08-02T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=88), bouteille=Bouteille(id=BouteilleId(idBouteille=1999, idRegion=2, idCouleur=3), nom=Vin ENI Ecole))]

find using query: { "client.quantity\_ordered" : { "$gt" : 100}} fields: Document{{}} for class: class fr.eni.gestionavis.bo.Avis in collection: reviews

[

Avis(id=64be818557e18425fae61855, note=3, commentaire=Commentaire (3), date=2023-07-23T15:28, client=Client(pseudo=patricketoile@email.fr, quantiteCommandee=108), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service)),

Avis(id=64be818557e18425fae61856, note=3, commentaire=Commentaire (3), date=2023-08-02T15:28, client=Client(pseudo=patricketoile@email.fr, quantiteCommandee=324), bouteille=Bouteille(id=BouteilleId(idBouteille=1999, idRegion=2, idCouleur=3), nom=Vin ENI Ecole)),

Avis(id=64be818557e18425fae61858, note=4, commentaire=Commentaire (4), date=2023-07-23T15:28, client=Client(pseudo=carlotentacule@email.fr, quantiteCommandee=400), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service)),

Avis(id=64be818557e18425fae61859, note=4, commentaire=Commentaire (4), date=2023-08-02T15:28, client=Client(pseudo=carlotentacule@email.fr, quantiteCommandee=1600), bouteille=Bouteille(id=BouteilleId(idBouteille=1999, idRegion=2, idCouleur=3), nom=Vin ENI Ecole))]

find using query: { "date" : { "$gt" : { "$date" : "2023-07-13T10:28:00Z"}, "$lt" : { "$date" : "2023-07-31T13:28:00Z"}}} fields: Document{{}} for class: class fr.eni.gestionavis.bo.Avis in collection: reviews

[

Avis(id=64be818557e18425fae61851, note=2, commentaire=Commentaire (2), date=2023-07-13T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=22), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61852, note=2, commentaire=Commentaire (2), date=2023-07-23T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=44), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service)),

Avis(id=64be818557e18425fae61854, note=3, commentaire=Commentaire (3), date=2023-07-13T15:28, client=Client(pseudo=patricketoile@email.fr, quantiteCommandee=36), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61855, note=3, commentaire=Commentaire (3), date=2023-07-23T15:28, client=Client(pseudo=patricketoile@email.fr, quantiteCommandee=108), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service)),

Avis(id=64be818557e18425fae61857, note=4, commentaire=Commentaire (4), date=2023-07-13T15:28, client=Client(pseudo=carlotentacule@email.fr, quantiteCommandee=100), bouteille=Bouteille(id=BouteilleId(idBouteille=18298, idRegion=3, idCouleur=1), nom=Vin ENI Edition)),

Avis(id=64be818557e18425fae61858, note=4, commentaire=Commentaire (4), date=2023-07-23T15:28, client=Client(pseudo=carlotentacule@email.fr, quantiteCommandee=400), bouteille=Bouteille(id=BouteilleId(idBouteille=1298, idRegion=3, idCouleur=2), nom=Vin ENI Service))]

* Les seuls non présents, sont ceux d’Aout 2023