Association – Référence manuelle

# Démonstration 8 du module 3

|  |
| --- |
| L’objectif de cette démonstration :   * Création d’une association manuelle entre 2 Documents |

Déroulement

# Contexte

Nous voulons une deuxième application, qui permettra de gérer les avis des stagiaires.

Elle permettra plus tard, de réaliser des statistiques sur les cours.

Notre application va se compléter au fil des démonstrations de ce module.

Elle est indépendante de l’application précédente.

Comme dans l’application précédente, les informations en base sont en anglais pour différencier les BO et les Document

A l’étape précédente, nous avons créé l’association : un stagiaire donne un avis

* Voici le schéma des classes déjà réalisé

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

Description générée automatiquement

* L’ensemble des 2 classes représente un Document
  + Et ils sont stockés dans la Collection : reviews
* La classe Formateur existe aussi, c’est un Document

Une image contenant texte, Police, blanc

Description générée automatiquement

* La note et le commentaire pédagogique se réfèrent à un formateur
  + Nous devons donc associer le Formateur correspondant
  + Voici le diagramme de classes à gérer :

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

Description générée automatiquement

* C’est une association (N:1)
  + N Avis peuvent être associés à 1 Formateur
  + Formateur étant déjà un Document, nous allons utiliser @DocumentReference pour l’associer
    - Il référence l’identifiant du Formateur en base

# Référence manuelle - Association (N:1)

## Modification de la classe BO : Avis

* Ajout de l’attribut Formateur
* Avec l’annotation @DocumentReference

**package** fr.eni.demo.bo;

**import** org.springframework.data.annotation.Id;

**import** org.springframework.data.mongodb.core.mapping.Document;

**import** org.springframework.data.mongodb.core.mapping.DocumentReference;

**import** org.springframework.data.mongodb.core.mapping.Field;

**import** lombok.\*;

@Data

@AllArgsConstructor

@NoArgsConstructor

@Builder

**public** **class** Avis {

@Id

**private** String id;

@Field(name = "pedagogical\_note")

**private** **int** notePedagogie;

@Field(name = "pedagogical\_commentary")

**private** String commentairePedagogie;

@Field(name = "course\_note")

**private** **int** noteCours;

@Field(name = "course\_commentary")

**private** String commentaireCours;

//Association

@Field("student")

**private** Stagiaire stagiaire;

//Association (N:1)

@DocumentReference

@Field("trainer\_id")

**private** Formateur formateur;

}

* + Ajout de l’annotation @Field pour préciser le nom du champ en base

## Création d’un test pour valider l’association

* Nous avons déjà un formateur en base, nous l’utiliserons pour le test
* Essayons d’insérer un Avis associé à au Formateur en base

**package** fr.eni.demo.bo.association;

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

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

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

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

**import** fr.eni.demo.bo.\*;

**import** fr.eni.demo.dal.\*;

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

@Slf4j

@SpringBootTest

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

**class** TestAssociationAvisFormateur {

@Autowired

AvisRepository avisRepository;

@Autowired

FormateurRepository formateurRepository;

**final** String emailFormateur = "abaille@campus-eni.fr";

@Test

**void** test01\_save\_avis\_formateur() {

**final** Formateur formateur = Formateur

.*builder*()

.email(emailFormateur)

.build();

**final** Avis avis = Avis

.*builder*()

.notePedagogie(4)

.commentairePedagogie("commentaire sur la pédagogie")

.noteCours(3)

.commentaireCours("commentaire sur le cours")

.stagiaire(Stagiaire

.*builder*()

.immatriculation("ENI\_CAMPUS\_202312456")

.promotion("EDMW0002")

.build())

.build();

//Association avec Formateur

avis.setFormateur(formateur);

//Sauver

**final** Avis avisDB = avisRepository.save(avis);

// Vérifier que l'identifiant n'est pas nul

*assertThat*(avisDB.getId()).isNotNull();

*assertThat*(avisDB.getId()).isNotBlank();

// Vérifier que l'identifiant du Formateur n'est pas nul

*assertThat*(avisDB.getFormateur().getEmail()).isNotNull();

**final** String emailDB = avisDB.getFormateur().getEmail();

*assertThat*(emailDB).isEqualTo(formateur.getEmail());

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

//Remarquer que seul l'identifiant du Formateur est présent. Les autres attributs sont nuls

// Pour récupérer les données du Formateur -> il faudrait une seconde requête

}

}

* Remarques :
  + Si une requête avait été appelé dans le test pour remonter le Formateur
  + Alors le contexte de Spring aurait fait que les données du Formateur soient présentes pour l’Avis

### Traces dans la console

* Spring Data MongoDB a bien inséré le Formateur dans la Collection reviews :

Inserting Document containing fields: [pedagogical\_note, pedagogical\_commentary, course\_note, course\_commentary, student, trainer\_id, \_class] in collection: reviews

* L’objet stocké en base permet bien de remonter une instance d’un Avis et l’identifiant du Formateur associé
  + Voici l’Avis enregistré :

Avis(id=64b93e1c8d6a1e5e120e9a2f, notePedagogie=4, commentairePedagogie=commentaire sur la pédagogie, noteCours=3, commentaireCours=commentaire sur le cours, stagiaire=Stagiaire(immatriculation=ENI\_CAMPUS\_202312456, promotion=EDMW0002), formateur=Formateur(email=abaille@campus-eni.fr, nom=null, prenom=null))

* + C’est là qu’on retrouve l’attribut formateur
  + Pour le moment, l’identifiant (email) est bien renseigné mais pas les autres attributs
  + Il faut une seconde requête pour cela
    - Ou avoir pré-chargé dans le contexte de Spring le Formateur en faisant un findById avant de l’associé à l’Avis
    - Ce qui correspond aussi à une seconde requête mais avant l’insertion

### En base :

* Et dans la Collection reviews
  + Il y a un nouvel Avis
  + Associé à un Formateur 🡪 référencé par son identifiant (email)



# Contraintes :

Attention, si vous ajouter un avis avec un identifiant quelconque. Il sera enregistré. Mais le Formateur n’existera pas en base.

Le seul cas, où il y aura une erreur c’est si vous mettez un identifiant nul.