Association – DBRef

# Démonstration 9 du module 3

|  |
| --- |
| L’objectif de cette démonstration :   * Création d’une association entre 2 Document avec @DBRef |

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 cette itération, nous allons gérer une association entre 2 Document dont l’un deux à un identifiant complexe Cours

* Les cours sont des données qui seront assez fixent et nombreuses.
* Il est donc intéressant de les gérer dans leur propre Collection
* Nous avons déjà la classe Cours. Elle est déclarée en un Document pour MongoDB
  + Avec un identifiant qui est une clef composite : CoursId

Avis est associé à son Formateur, il faut aussi l’associer à son Cours

* C’est une association (N:1)
  + N Avis peuvent être associés à 1 Cours
* Cours ayant un identifiant en clef composite
  + Il faut utiliser @DBRef pour référencer cette association
* Voici le diagramme des classes :

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

Description générée automatiquement

# Contexte en base

Avec le test de la clef composite, nous avons déjà enregistré un Cours en base

{

\_id: {

reference: 'M360',

computer\_science\_course: 'Développement'

},

title: 'Java Frameworks - API Web',

duration: 10,

\_class: 'fr.eni.demo.bo.clefcomposite.Cours'

}

# Référence manuelle - Association (N:1)

## Modification de la classe BO : Avis

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

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

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

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

**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** fr.eni.demo.bo.clefcomposite.Cours;

**import** lombok.\*;

@Data

@AllArgsConstructor

@NoArgsConstructor

@Builder

@Document(collection = "reviews")

**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;

//Association (N:1)

@DBRef

@Field("computer\_course\_id")

**private** Cours cours;

}

* + Ajout de l’annotation @Field pour préciser le nom des champs en base

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

* Essayons d’insérer un Avis et son Cours
  + Pour voir la différence sur le contexte de Spring
  + Nous allons charger le Cours depuis la base de données
  + Puis nous l’associerons avec l’Avis
  + Nous pourrons ainsi constater qu’il restera dans le contexte et pourra retourner tous ses champs

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

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

**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.demo.bo.\*;

**import** fr.eni.demo.bo.clefcomposite.Cours;

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

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

@Slf4j

@SpringBootTest

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

**class** TestAssociationAvisCours {

@Autowired

AvisRepository avisRepository;

@Autowired

CoursRepository coursRepository;

@Test

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

// Récupération des Cours en base

**final** List<Cours> listeCoursDB = coursRepository.findAll();

*assertThat*(listeCoursDB).isNotNull();

*assertThat*(listeCoursDB).isNotEmpty();

**final** Cours coursDB = listeCoursDB.get(0);

**final** Avis avis = Avis

.*builder*()

.notePedagogie(5)

.commentairePedagogie("commentaire sur la pédagogie")

.noteCours(4)

.commentaireCours("commentaire sur le cours")

.stagiaire(Stagiaire

.*builder*()

.immatriculation("ENI\_CAMPUS\_202311356")

.promotion("EDMW0001")

.build())

.build();

// Association avec Cours

avis.setCours(coursDB);

// 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 le Cours est complet

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

*assertThat*(avisDB.getCours()).isEqualTo(coursDB);

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

}

}

### Traces dans la console

* Spring Data MongoDB a bien inséré le stagiaire et son avis dans la collection students :

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

* + Il y a bien le champ du Cours associé
  + Remarquer : qu’il n’y a pas le champ Formateur 🡪 nous n’avons pas renseigné de Formateur
* Dans l’objet stocké en base permet bien de remonter l’instance du Cours associé à l’Avis
  + Et noteCours et commentaireCours

Avis(

id=64b943948a0f266b6c351fc5,

notePedagogie=5,

commentairePedagogie=commentaire sur la pédagogie,

noteCours=4,

commentaireCours=commentaire sur le cours,

stagiaire=Stagiaire(immatriculation=ENI\_CAMPUS\_202311356, promotion=EDMW0001),

formateur=null,

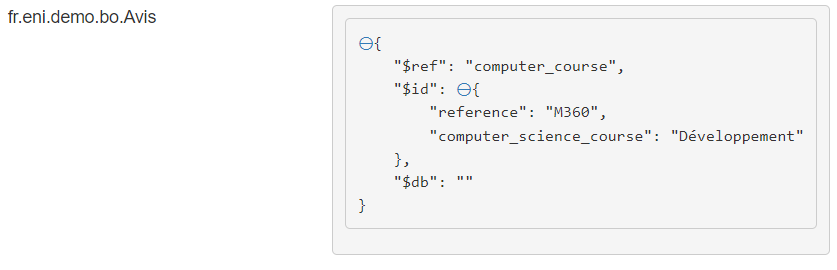
cours=Cours(id=CoursId(reference=M360, filiere=Développement), titre=Java Frameworks - API Web, duree=10)

)

### En base :



* Il y a bien un Document Avis qui contient
  + La référence vers un Cours via DBRef
* Sur la vue de la Collection ; on voit le détail de l’objet DBRef :



* Il y a bien les 3 champs :
  + $ref : "computer\_course"
  + $id : qui est le JSON CoursId
  + $db : qui n’est pas précisé (car optionnel) 🡪 Si vous voulez que $db soit préciser, il faut ajouter le paramètre à l’annotation
* Nous avons à présent
  + Une association intégrée, embarquée 🡪 Avis – Stagiaire 🡪Préconisée par MongoDB
  + Une association manuelle, par \_id 🡪 Avis – Formateur 🡪 @DocumentReference
  + Une association DBRef 🡪 Avis – Cours 🡪 @DBRef