Appel de requêtes spécifiques sur des associations

# Démonstration 04 du module 5

|  |
| --- |
| L’objectif de cette démonstration est   * Voir la contrainte avec les associations avec les requêtes spécifiques |

Contexte

* Poursuivre dans notre deuxième application, qui permettra de gérer les avis des stagiaires.

Déroulement

# Pour les méthodes avec paramètres complexes

* Attention, il ne faut pas essayer de passer un objet JSON en paramètre de l’URL

Nous avions créé des méthodes qui prenaient en argument un objet complexe : Stagiaire, Formateur ou Cours

C’est possible dans le cas d’un service web classique (@RestController). Dans ce cas, l’url comporte uniquement l’identifiant de l’entité. La couche service récupère l’entité avant d’appeler le Repository.

Mais dans le cas d’un Spring Data REST, il faut uniquement travailler sur les identifiants des associations et les attributs simples

## findByStagiaire

* Pour rechercher via l’association d’un Stagiaire.
* Il faut remplacer findByStagiaire par findByStagiaireImmatriculation
  + « immatriculation » est un champ unique (il correspond à une clef primaire pour MongoDB)

…

@RepositoryRestResource(collectionResourceRel = "avis", path = "avis")

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

List<Avis> findByNoteCours(@Param("noteCours") **int** noteCours);

List<Avis> findByNoteCoursGreaterThan(@Param("noteCours") **int** noteCours);

List<Avis> findByNoteCoursLessThan(@Param("noteCours") **int** noteCours);

// List<Avis> findByStagiaire(@Param("stagiaire") Stagiaire stagiaire);

List<Avis> findByStagiaireImmatriculation(@Param("immatriculation") String immatriculation);

…

}

* Tester avec http://localhost:8080/avis/search/findByStagiaireImmatriculation?immatriculation=ENI\_12531
* Modifier le test unitaire associé à la méthode findByStagiaire en appelant findByStagiaireImmatriculation

…

@Test

**void** test04\_findByStagiaire() {

**final** Stagiaire stagiaire = Stagiaire

.*builder*()

.immatriculation("ENI\_12531")

.promotion("CDA12341")

.build();

List<Avis> listeAvis = avisRepository.findByStagiaireImmatriculation(stagiaire.getImmatriculation());

*assertThat*(listeAvis).isNotNull();

*assertThat*(listeAvis).isNotEmpty();

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

***log***.info("Nb Avis du Stagiaire (" + stagiaire.toString() + ") : " + listeAvis.size());

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

}

…

* + Le test recompile et est de nouveau opérationnel

## findByFormateur

* L’association est similaire à la précédente, l’identifiant de Formateur est son email
* Remplacer findByFormateur par findByFormateurEmail

@RepositoryRestResource(collectionResourceRel = "avis", path = "avis")

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

List<Avis> findByNoteCours(@Param("noteCours") **int** noteCours);

List<Avis> findByNoteCoursGreaterThan(@Param("noteCours") **int** noteCours);

List<Avis> findByNoteCoursLessThan(@Param("noteCours") **int** noteCours);

// List<Avis> findByStagiaire(@Param("stagiaire") Stagiaire stagiaire);

List<Avis> findByStagiaireImmatriculation(@Param("immatriculation") String immatriculation);

// List<Avis> findByFormateur(@Param("f") Formateur f);

List<Avis> findByFormateurEmail(@Param("email") String email);

…

}

* Tester avec http://localhost:8080/avis/search/findByFormateurEmail?email=pmontembault@campus-eni.fr
* Modifier le test unitaire associé à la méthode findByFormateur en appelant findByFormateurEmail

…

@Test

**void** test05\_findByFormateur() {

**final** Formateur philippe = Formateur

.*builder*()

.email("pmontembault@campus-eni.fr")

.nom("MONTEMBAULT")

.prenom("Philippe")

.build();

List<Avis> listeAvis = avisRepository.findByFormateurEmail(philippe.getEmail());

*assertThat*(listeAvis).isNotNull();

*assertThat*(listeAvis).isNotEmpty();

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

***log***.info("Nb Avis sur le Formateur (" + philippe.toString() + ") : " + listeAvis.size());

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

}

…

* + Le test recompile et est de nouveau opérationnel