Gestion des avis (Partie 10)

# TP01 du module 09 – AOP

|  |
| --- |
| Ce TP sera un TP fil rouge du cours   * Utilisation de l’AOP * Ajout des starters et de la configuration |

|  |
| --- |
| **Durée estimée** |
| 45 minutes |

Contexte

Dans les itérations de ce module, nous allons ajouter des aspects.

* Si vous n’aviez pas fini le TP du module 08. Vous trouverez les classes solutions dans un zip des ressources.
* Attention, la configuration Spring Boot n’est pas incluse
* Ainsi que les propriétés pour gérer OAuth 2.0 dans le fichier de configuration de Spring Boot : application.yml

Solution

# Ajout du starter de Spring AOP

// Starter AOP

**implementation** 'org.springframework.boot:spring-boot-starter-aop:3.1.3'

* Faire un Gradle Refresh Project
* Votre projet permet de créer des aspects avec Spring AOP

# Gestion d’aspects

* Modifier le niveau des logs générales à warning
* Nous allons manipuler plus de logs qu’au paravent.
  + Pour éviter de saturer les ressources, nous allons changer le niveau de trace du projet
* Dans le fichier de configuration de Spring Boot : application.yml :
  + Nous avions un niveau de log pour MongoTemplate
  + Nous ajoutons un niveau pour root : WARN

#Log

logging:

level:

root: WARN

org:

springframework:

data:

mongodb:

core:

MongoTemplate: DEBUG

# Gestion d’aspects

* Pour montrer que les aspects sont très indépendants du code
* Copier votre classe de démonstration dans le package fr.eni.gestionavis.aop
  + Retirer les méthodes sur les advices de type after
* Et modifier le pointcut de l’advice before comme celui du around

**package** fr.eni.gestionavis.aop;

**import** java.util.List;

**import** org.aspectj.lang.\*;

**import** org.aspectj.lang.annotation.\*;

**import** org.springframework.stereotype.Component;

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

@Aspect

@Component

@Slf4j

**public** **class** LogAspect {

@Before("execution(public \* \*..AvisRepository.findByNote\*(int))")

**public** **void** before(JoinPoint jp) {

***log***.warn("ENI AOP - @Before - " + jp.getSignature().getName());

}

@Around(value = "execution(public \* \*..AvisRepository.findByNote\*(..))")

**public** Object around(ProceedingJoinPoint call) **throws** Throwable {

Object result = **null**;

**long** before = System.*currentTimeMillis*();

result = call.proceed();

**long** after = System.*currentTimeMillis*();

**long** duration = after - before;

String nomMethode = call.getSignature().getDeclaringTypeName() + "/"

+ call.getSignature().getName();

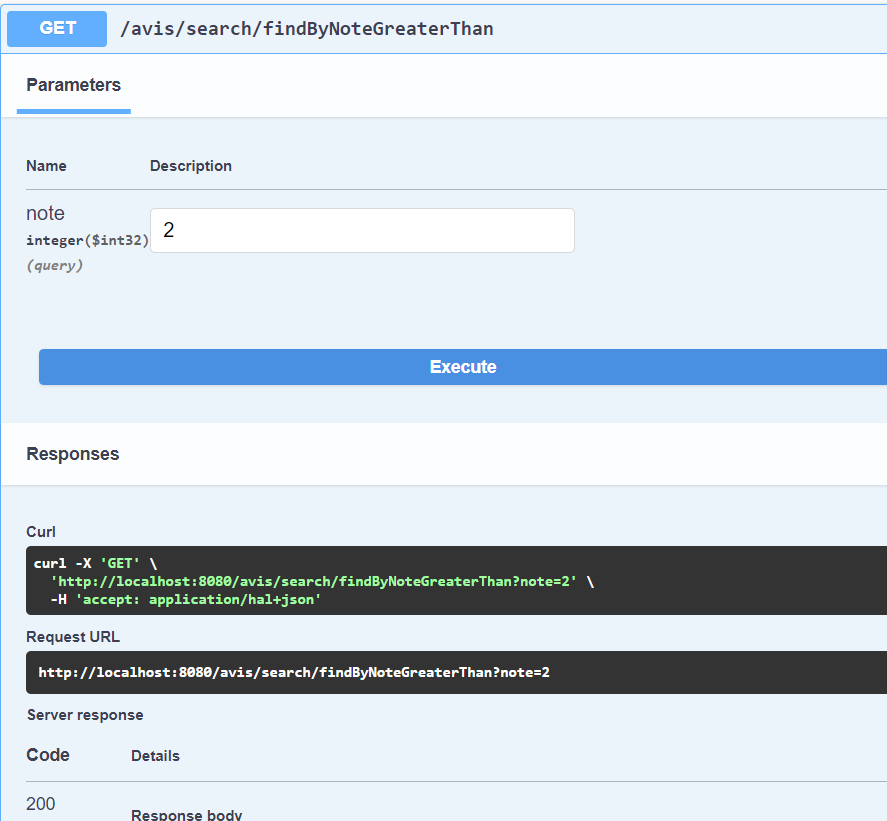
***log***.warn("ENI AOP - @Around - " + nomMethode + ":" + duration + " ms");

**return** result;

}

}

* L’inconvénient du « pointcut » du « advice before » est qu’il utilisait les packages explicites
  + En ne précisant plus les packages, le même pointcut peut fonctionner dans tout projet contenant une classe ou interface AvisRepository et déclarant des méthodes avec un nom commençant par findByNote
* Plus vous serez généraliste sur votre pointcut plus vous pourrez rendre réutilisable un aspect
* Exécuter et constater que les 2 aspects s’appliquent



fr.eni.gestionavis.aop.LogAspect ENI AOP - @Before - findByNoteGreaterThan

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

fr.eni.gestionavis.aop.LogAspect ENI AOP - @Around - fr.eni.gestionavis.dal.AvisRepository/findByNoteGreaterThan:86 ms

# Ajout d’un aspect

* Créer un aspect pour tracer les 2 méthodes associées au client :
  + findByClientPseudo
  + findByClientQuantiteCommandeeGreaterThan
* Il faut tracer la liste des Avis retournées et le nom de la méthode
* Il faut un advice after returning
  + Il s’exécute si la méthode arrive à retourner son résultat
  + C’est équivalent à un return dans le bloc TRY de TRY-CATCH-FINALLY
* Ajouter une méthode, elle prend en paramètre JointPoint et la liste des Avis retournée
  + Utiliser @AfterReturning et paramétré « returning »

@AfterReturning(pointcut = "execution(public \* \*..AvisRepository.findByNote\*(..))", returning = "result")

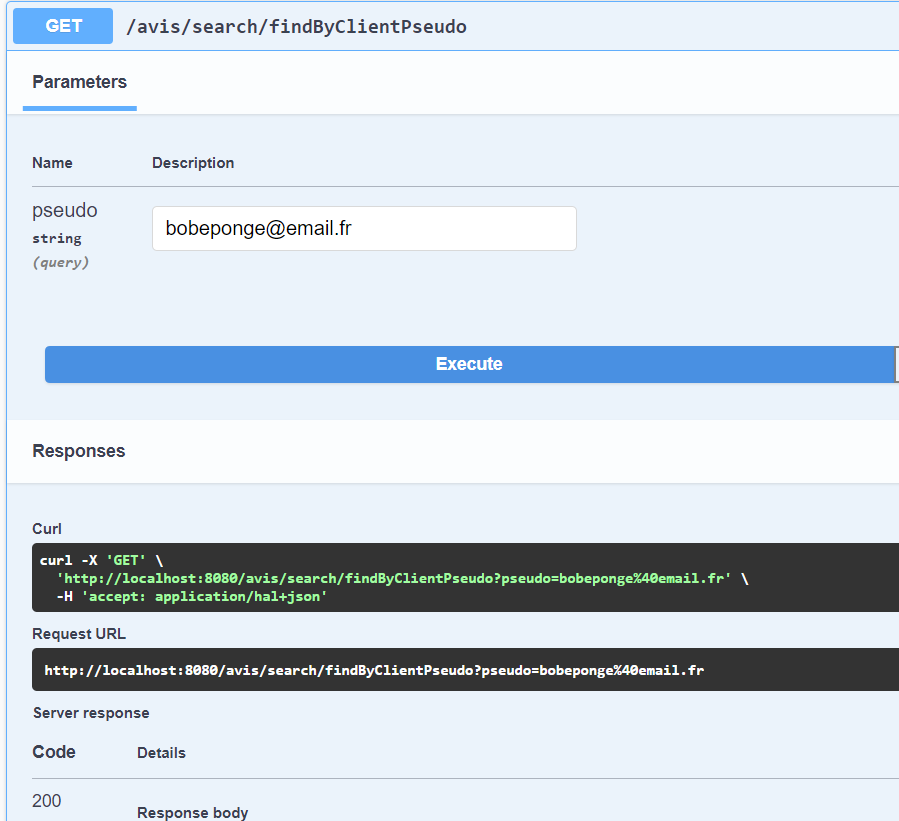
**public** **void** afterReturning(JoinPoint jp, List<Avis> result) {

***log***.warn("ENI AOP - @AfterReturning - TRY - RETURN [" + jp.getSignature().getName() + "] = " + result);

}

### Exécution :

* Tester l’appel des 2 méthodes



* Les logs seront similaires à :

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

fr.eni.gestionavis.aop.LogAspect ENI AOP - @AfterReturning - TRY - RETURN [findByClientPseudo] = [Avis(id=64f8754fed97c85863f6fc38, note=2, commentaire=Commentaire (2), date=2023-07-13T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=22), bouteille=Bouteille(idBouteille=18298, idRegion=3, idCouleur=1, nom=Vin ENI Edition)), Avis(id=64f8754fed97c85863f6fc39, note=2, commentaire=Commentaire (2), date=2023-07-23T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=44), bouteille=Bouteille(idBouteille=1298, idRegion=3, idCouleur=2, nom=Vin ENI Service)), Avis(id=64f8754fed97c85863f6fc3a, note=2, commentaire=Commentaire (2), date=2023-08-02T15:28, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=88), bouteille=Bouteille(idBouteille=1999, idRegion=2, idCouleur=3, nom=Vin ENI Ecole)), Avis(id=64f8754fed97c85863f6fc42, note=4, commentaire=Doux. A déguster frais, date=2023-09-06T14:49:19.625, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=11), bouteille=null), Avis(id=64f8754fed97c85863f6fc43, note=5, commentaire=Doux. A déguster frais, date=2023-09-06T14:49:19.641, client=Client(pseudo=bobeponge@email.fr, quantiteCommandee=11), bouteille=Bouteille(idBouteille=18298, idRegion=3, idCouleur=1, nom=Vin ENI Edition))]

* + La requête de MongoTemplate
  + Et l’aspect @AfterReturning qui retourne la liste des avis