

TP – Services Avancés

PLAN DU COURS

- Affectations
- Désaffectation
- TP

Affectation

Dans le même projet Spring Boot (kaddem) déjà créé, Coder le CrudRepository, le Service (l'Interface) et le Controller de l'Entité **Projet**. Inspirez-vous du code déjà fait pour l'Entité ProjetDetail.

Affectation 1

Exercice 1 : Créer le service qui permet d'ajouter un Projet et un Projet Detail et d'affecter le ProjetDetail au Projet

Solution :

- Dans l'entité Projet (Parent) on doit cascader les actions du Projet vers le projetDetail
- Dans la méthode, on doit faire un ajout simple :

Affectation 1

Service :

```
public Projet addProjetAndProjetDetailAndAssign(Projet projet) {  
    return projetRepository.save(projet);  
}
```

Controller à finir pour avoir l'URL :

<http://localhost:8089/kaddem/projet/ajouter-projet-et-projet-detail>

Objet JSON :

```
{  
  "sujet": "projet numéro 1",  
  "projetDetail": {  
    "description": "détail du projet",  
    "technologie": "Java",  
    "cout_provisoire": 1000,  
    "dateDebut": "2022-11-14T23:52:48.343Z"  
  }  
}
```

Affectation 1

Table t_projet_detail :

pd_id	pd_cout_provisoire	date_debut	pd_description	pd_technologie
16	0	2022-11-15	string	string

Table t_projet :

projet_id	projet_sujet	projet_detail_pd_id
2	string	16

Affectation 2

Exercice 2 : Créer le service qui permet d'Affecter un ProjetDetail à un Projet (le ProjetDetail et le Projet sont déjà créés dans la base de données :

Solution :

Dans la méthode, on doit récupérer de la base de données le Projet et le ProjetDetail, puis affecter le ProjetDetail (fils) au Projet (parent), puis faire une mise à jour :

Affectation 2

Service :

```
public void assignProjetDetailToProjet(Long projetId, Long projetDetailId) {  
    Projet projet = projetRepository.findById(projetId).get();  
    ProjetDetail projetDetail = projetDetailRepository.findById(projetDetailId).get();  
    // on set le fils dans le parent :  
    projet.setProjetDetail(projetDetail);  
    projetRepository.save(projet);  
}
```

Controller à finir pour avoir l'URL :

<http://localhost:8089/kaddem/projet/affecter-projet-detail-a-projet/1/1>

Affectation 3

Exercice 3 : Créer le service qui permet d'Affecter un Projet à une Equipe (le Projet et l'Equipe sont déjà créés dans la base de données) :

Solution :

Dans la méthode, on doit récupérer de la base de données l'Equipe et le Projet, puis affecter le Projet (fils) à l'Equipe (parent), puis faire une mise à jour :

Affectation 3

Service :

```
public void assignProjetToEquipe(Long projetId, Long equipId) {  
    Projet projet = projetRepository.findById(projetId).get();  
    Equipe equipe = equipeRepository.findById(equipId).get();  
    // on set le fils dans le parent :  
    equipe.getProjets().add(projet);  
    equipeRepository.save(equipe);  
}
```

Controller à finir pour avoir l'URL :

<http://localhost:8089/kaddem/projet/affecter-projet-equipe/1/1>

Affectation 4

Exercice 4 : Créer le service qui permet d'Ajouter un Projet d'Affecter un Projet Detail (déjà dans la base de données) à ce projet :

Solution :

Dans la méthode, on doit récupérer de la base de données le Projet Detail, puis le setter dans la le Projet (parent), puis faire une mise à jour :

Affectation 4

Service :

```
public Projet addProjetAndAssignProjetToProjetDetail(Projet projet, Long projetDetailId) {  
    ProjetDetail projetDetail = projetDetailRepository.findById(projetDetailId).get();  
    // on set le fils dans le parent :  
    projet.setProjetDetail(projetDetail);  
    return projetRepository.save(projet);  
}
```

Controller à finir pour avoir l'URL :

<http://localhost:8089/kaddem/projet/affecter-projet-equipe/1>

(le projet sera envoyé dans le body, l'id du projetdetail dans le path).

Désaffectation 1

Exercice 5 : Créer le service qui permet de désaffecter un ProjetDetail d'un Projet (Le Projet et le ProjetDetail sont déjà créés dans la base de données) :

Solution :

Récupérer le Projet de la base de données. Puis, il suffit de setter l'attribut projetDetail à **null** dans Projet, puis faire une mise à jour :

Désaffectation 1

Service :

```
public Projet DesaffecterProjetDetailFromProjet(Long projetId) {  
    Projet projet = projetRepository.findById(projetId).get();  
    projet.setProjetDetail(null);  
    return projetRepository.save(projet);  
}
```

Controller à finir pour avoir l'URL :

<http://localhost:8089/kaddem/projet/desaffecter-projet-detail/1>

Désaffectation 1

Table t_projet_detail :

pd_id	pd_cout_provisoire	date_debut	pd_description	pd_technologie
16	0	2022-11-15	string	string

Table t_projet :

projet_id	projet_sujet	projet_detail_pd_id
2	string	

Désaffectation 2

Exercice 5 : Créer le service qui permet de désaffecter un Peojet d'une Equipe (L'Equipe et le Projet sont déjà créés dans la base de données) :

Solution :

Récupérer le Projet et l'Equipe de la base de données. Puis, il suffit d'enlever le projet à l'équipe, puis faire une mise à jour :

Désaffectation 2

Service :

```
public void desaffecterProjetFromEquipe(Long projetId, Long equipId) {  
    Projet projet = projetRepository.findById(projetId).get();  
    Equipe equipe = equipeRepository.findById(equipId).get();  
    // on enlève le fils du parent :  
    equipe.getProjets().remove(projet);  
    equipeRepository.save(equipe);  
}
```

Controller à finir pour avoir l'URL :

<http://localhost:8089/kaddem/projet/desaffecter-projet-detail/1/1>

TP1

Implémenter l'ensemble des méthodes (**de page 3 à page 17**) ci-dessous et les tester avec Postman et Swagger. **(Homework)**

Attention : Pour pouvoir faire les affectation de désaffectation, on doit injecter les repository nécessaires dans le service ProjetServiceImpl :

```
@Service
public class ProjetServiceImpl implements IProjetService {
    @Autowired
    ProjetRepository projetRepository;
    @Autowired
    ProjetDetailRepository projetDetailRepository;
    @Autowired
    EquipeRepository equipeRepository;
```

TP2

Reprendre le TP (TP en Classe) et implémenter les méthodes d'Ajout, d'Affectation et de Désaffectation et les tester avec Postman et Swagger. **(Ce sera fait en classe la séance prochaine).**