

Durant le projet E-Music j'ai utilisé le framework spring boot.

Compétences utilisés :

- Utilisation de composants d'accès aux données
- Exploitation des ressources du cadre applicatif (framework)

Pour la compétence composants d'accès aux données nous avons donc utilisé le module du framework Spring Data JPA qui permet d'interagir avec une base de données.

Voici un exemple de son implémentation :

```
4 usages  KilianL55
public interface IRepoEnfant extends CrudRepository<Enfant, Integer> {

    1 usage  KilianL55
    public Enfant findByUsername(String username);
    1 usage  KilianL55
    public List<Enfant> findByResponsable(Responsable responsable);
    1 usage  KilianL55
    public Enfant findByUsernameAndResponsable(String username, Responsable responsable);
}
```

Voici l'implémentation d'une interface JPA, cette interface permet donc d'accéder aux données grâce au fait que l'on a étendu **CrudRepository** à l'interfac.

Voici comment injecter le repository dans le code :

```
12 usages
@Autowired
private IRepoResponsable repoResponsable;
```

Voici l'exemple d'une lecture dans la base de données :

Le but est de trouver tous les enfants qui appartiennent au responsable.

```
Responsable parent = (Responsable) responsable;
List<Enfant> enfants = repoEnfant.findByResponsable(parent);
```

Voici l'exemple de mettre à jour des données dans la base de données :

```

KilianL55 +1
@PostMapping("/updateResponsable")
public RedirectView updateResponsable( @ModelAttribute Responsable responsable){
    Responsable resp = (Responsable) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    resp.setNom(responsable.getNom());
    resp.setPrenom(responsable.getPrenom());
    resp.setUsername(responsable.getUsername());
    resp.setAdresse(responsable.getAdresse());
    resp.setAdresse2(responsable.getAdresse2());
    resp.setVille(responsable.getVille());
    resp.setCode_postal(responsable.getCode_postal());
    resp.setQuotient_familial(responsable.getQuotient_familial());
    resp.setTel1(responsable.getTel1());
    resp.setTel2(responsable.getTel2());
    resp.setTel3(responsable.getTel3());
    repoResponsable.save(resp);
    return new RedirectView( url: "dashboard");
}

```

Pour mettre à jour les données de l'utilisateur connecté, on récupère cet utilisateur grâce aux modules webSecurity, puis on met à jour les données de cet utilisateur grâce à nos méthodes set, puis on les sauvegarde grâce à JPA via notre repository.

Pour plus d'exemple consultez le repository Github du projet e-music :

<https://github.com/KilianL55/E-Music>