TP3 : Filtre, Pagination, javascript

# Fonctionnement

Avant de Commencer

1. Créez un repository pour le TP3
2. Copiez le projet de départ dans le répertoire de votre repository

Pour chaque partie

1. Créez une branche appelée **FCT\_PartieX** où X est le numéro de la partie
2. Assurez-vous d'être dans la branche que vous avez créée dans LOCAL
3. À la fin de la partie, faire un merge de la branche dans main avant de commencer la partie suivante.

Fonctionnement

1. Dans ce TP, nous vous demandons de mettre en place certaines fonctionnalités de javascript. (JQuery, AJAX) appliquer à la vue index de Trainer. D’éventuelle mise à jour pour **TrainerControlleur** sera demandée selon le besoin.

# Partie 1 : Filtre (20 Pts)

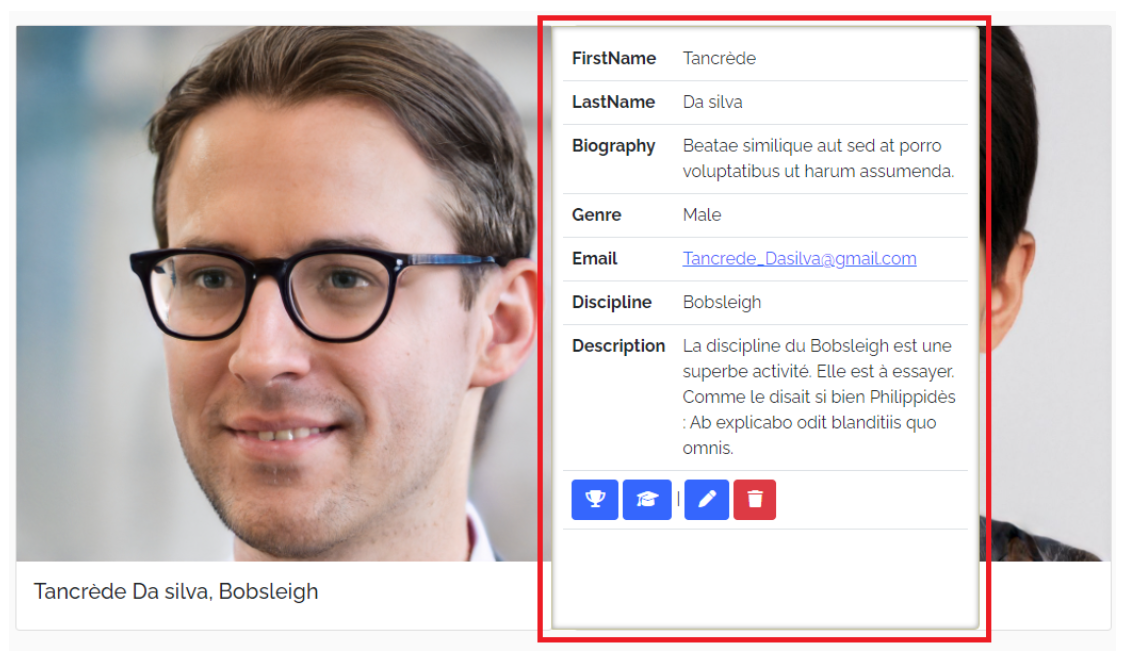
1. Implémentez la fonction **Filter** du **TrainerController** pour retourner une vue Index avec le filtre comme modèle.
2. Compléter la méthode **GetAllAsync** du service **TrainerService** pour que le filtre prend en compte les critères ( **Discipline, Gender**)
   * Ajouter des **Where** pour filtrer selon chacun des critères ajoutés (**Discipline, Gender**)

**Exemple**

## Voici comment faire une requête pour une condition optionnelle dans Linq et ne filtrer que lorsque l'on sélectionne une valeur dans le filtre.

.Where(x=> filter.SelectedCategoryId == null || x.CategoryId == filter.SelectedCategoryId )

# Partie 2 : Javascript : Affichez le détail d'un Trainer dans la page Trainer/Index 35 Pts)

1. Écrivez du javascript en utilisant **jQuery** pour ajouter la classe **show** à l’élément enfant **aside** lorsqu'on survole le **card** d’un entraîneur.
2. Écrivez du javascript en utilisant **jQuery** pour retirer la classe **show** à l’élément enfant **aside** lorsqu'on ne survole plus le **card**.

# Partie 3 :JQUERY Pour Pagination (25 Pts)

## Faire fonctionner la pagination[​](https://info.cegepmontpetit.ca/3W6-Web-Transactionelle/tp/tp2#faire-fonctionner-la-pagination)

1. Dans cette partie la pagination s’affiche mais elle n’est pas fonctionnelle.

Image Reference

1. Ajoutez le javascript en utilisant **jQuery** pour que la page change dans le filtre du formulaire et qu'il se soumette automatiquement lorsqu'on clique sur un élément de pagination (incluant Previous et Next). Utilisez les attributs **data-page-id** qui sont déjà là sur les éléments de navigation. Voici les étapes nécessaires en JS :
   * Exécuter une fonction JS lorsqu'un élément de pagination est **cliqué**.
   * Obtenir la valeur du **data-page-id** de l'élément sur lequel l'utilisateur a cliqué.
   * Image ReferenceObtenir l'élément qui contient l'information du **SeletectedPageIndex.**
   * Modifier la valeur de l'élément en question

ATTENTION

C'est le moment de **tester**, ajouter des **commentaires** et faire votre **commit** et votre **push**

# Partie 4 : ajax : Ajouter un trainer comme favori Trainer (20 Pts)

* L’idée est d’identifier un **Trainer** favori en cliquant sur une icône

## **Avant de commencer**: désactiver le scripte qui permet d’affichez le détail d'un Trainer dans la page Trainer/Index

* **Contrôleur TrainerController** :
  + Ajouter une méthode **AddToFavorites (int ? id)**
  + La méthode **AddToFavorites** utilise le service pour mettre à jour **Trainer**, y compris la propriété Active.
* **Vue index.cshtml et JavaScript :**
  + La vue affiche la liste des entraîneurs avec la possibilité de les marquer comme favoris.
  + Le script JavaScript gère l'événement de clic pour effectuer un appel AJAX afin de mettre à jour l'état de favori de l'entraîneur.

**Fin du TP3!**