

PlaylistRepository.php :

```
/**
 * Retourne toutes les playlists triées sur le nombre de formations associées.
 *
 * @param string $ordre 'ASC' ou 'DESC'
 *
 * @return Playlist[]
 */
public function findAllOrderByNbFormations(string $ordre): array
{
    if (!in_array($ordre, [self::ORDER_ASC, self::ORDER_DESC], true)) {
        $ordre = self::ORDER_ASC;
    }

    return $this->createQueryBuilder('p')
        ->leftJoin('p.formations', 'f')
        ->addSelect('COUNT(f.id) AS HIDDEN nbFormations')
        ->groupBy('p.id')
        ->orderBy('nbFormations', $ordre)
        ->getQuery()
        ->getResult();
}
```

Modification effectuée :

Cette méthode que j'ai ajoutée permet de faire le compte des formations par playlist (COUNT(f.id)) et tri asc/desc sur le nombre de formations.

PlaylistController.php :

```
case 'nbformations':
    $playlists = $playlistRepository->findAllOrderByNbFormations($ordre);
    break;
```

Modification effectuée :

Ici j'ai fait en sorte que la nouvelle méthode soit appelée et que lorsque l'on utilise sur les flèches de tri ça filtre la nouvelle colonne.

Playlists.html.twig :

```
<th class="text-center align-top" scope="col">
  nb formations<br />
  <a href="{{ path('playlists.sort', {champ:'nbformations', ordre:'ASC'}) }}"
    class="btn btn-info btn-sm active" role="button" aria-pressed="true"></a>
  <a href="{{ path('playlists.sort', {champ:'nbformations', ordre:'DESC'}) }}"
    class="btn btn-info btn-sm active" role="button" aria-pressed="true"></a>
</th>
```

Modification effectuée :

J'ai créé une nouvelle colonne avec deux boutons de tri <> et appelé grâce au contrôleur « champ : 'nbformations' »

```
<td class="text-center">
  {{ playlists[k].formations|length }}
</td>
```

Modification effectuée :

Ici j'ai ajouté pour afficher le nombre réel de formations dans chaque playlist.

Playlist.html.twig :

```
<strong>nombre de formations :</strong>{{ playlistformations|length }}
```

Modification effectuée :

Là ça affiche le nombre de formations sur la page de détail d'une playlist.

Résultat :

Décroissant :

playlist	nb formations	catégories	
<div><div>< ></div><div><input type="text"/></div><div>filtrer</div></div>	<div><div>< ></div></div>	<div><input type="text"/></div>	
Bases de la programmation (C#)	74	C# POO	Voir détail
Programmation sous Python	19	Python POO	Voir détail
MCD : exercices progressifs	18	MCD	Voir détail
TP Android (programmation mobile)	18	Android SQL Java	Voir détail
Compléments Android (programmation mobile)	13	Android	Voir détail
Visual Studio 2019 et C#	11	C# POO	Voir détail
Cours UML	10	UML Cours	Voir détail
Eclipse et Java	8	Java UML	Voir détail
Exercices objet (sujets EDC BTS SIO)	8	POO	Voir détail
MCD exercices d'examen (sujets EDC BTS SIO)	8	MCD	Voir détail

Croissant :

playlist	nb formations	catégories	
<div><div>< ></div><div><input type="text"/></div><div>filtrer</div></div>	<div><div>< ></div></div>	<div><input type="text"/></div>	
playlist test	0		Voir détail
Cours Informatique embarquée	1	Cours	Voir détail
Cours Merise/2	1	MCD Cours	Voir détail
Cours Modèle relationnel et MCD	1	MCD Cours	Voir détail
Cours de programmation objet	1	POO Cours	Voir détail
Cours Composant logiciel	2	Cours	Voir détail
Cours MCD MLD MPD	2	MCD Cours	Voir détail
Cours MCD vs Diagramme de classes	2	MCD Cours	Voir détail
Cours Curseurs	2	SQL Cours POO	Voir détail

Sur la page de playlist :

Bases de la programmation (C#)

catégories : C# POO

description :

Exemples progressifs de programmes en procédural, événementiel et objet sous Visual Studio (version Entreprise 2017).

Prérequis : aucun

1ère partie : programmation procédurale en mode console (non graphique)

n°1 à 30 : procédural, notions élémentaires (variables, saisie/affichage, affectations/calculs, alternatives (if/switch), itérations (while/do-while/for))

n°31 à 42 : procédural, tableaux (1 et 2 dimensions, manipulations, tris, recherches)

n°43 à 59 : procédural, modules et paramètres (procédures et fonctions)

2ème partie : événementiel (en mode graphique)

n°60 à 67 : événementiel (programmation graphique)

3ème partie : initiation à l'objet

n°68 à 74 : notions de base en programmation objet sur des classes "métier"

nombre de formations :74

Nombre de formation présente.