

Exercices supplémentaires

Pour cet exercice vous pouvez récupérer les sources sur Github à l'adresse suivante :

promo05

Vous allez gérer dans cette partie la lecture des morceaux de musique.

Utilisez le type suivant pour le status de la lecture d'un album de musique :

```
// album.ts

export class StatusPlayer {
  albumId : string;
  song : string;
  playing: boolean = false;
  current: number;
  total: number;
  ratio: number = 0;
}
```

Implémentez la fonctionnalité suivante :

- 1/ Lorsque la lecture des morceaux d'un album est lancée la lecture des autres albums est impossible. Le bouton "play" est alors désactivé. Ce dernier se réactivera une fois que la lecture des morceaux d'un album est terminé.

Refactoring

Vous allez maintenant refactorer le component statistic.component.ts pour qu'il fonctionne avec Firebase.

Puis créez un pipe numberSongs, il utilisera le service AlbumService et une méthode getAlbumListNumber(id : string) qui retournera le nombre de chanson(s) pour un album donné.

La particularité de ce pipe c'est qu'il retournera un observable. Dans le script du template vous appellerez ce pipe de la manière suivante :

```
html <th >{{ (album.id | numberSongs) | async }}</th>
```

(*) ordre par rapport à la moyenne

Ordonné par ordre croissant (*)

Ordonné par ordre décroissant (*)

ref	Title	Song(s)	Average
Hardu	Kirby Ortega	5	15.4
Rock	Traci Huber	4	15.4
Jazzz	Dickerson Maynard	3	14.5
Fusio	Lindse	7	14.5

Previous 1 2 3 Next

Figure 1: Statistic