

Pagination

Vous allez mettre en place la pagination dans l'application. Ce TP n'est pas trop détaillé, il vous permettra de valider vos compétences vues sur la semaine.

Nous indiquons ce que nous souhaitons faire, vous êtes cependant libre d'avoir une approche personnelle de la problématique.

Nous vous donnons quelques conseils pour vous guider dans la suite du document.

Exercice 21

Créez un component propre à la pagination.

`ng g c paginate`

Mettez en place le component paginate puis placez le code HTLM de la manière suivante dans l'application (utilisez le Bootstrap) :

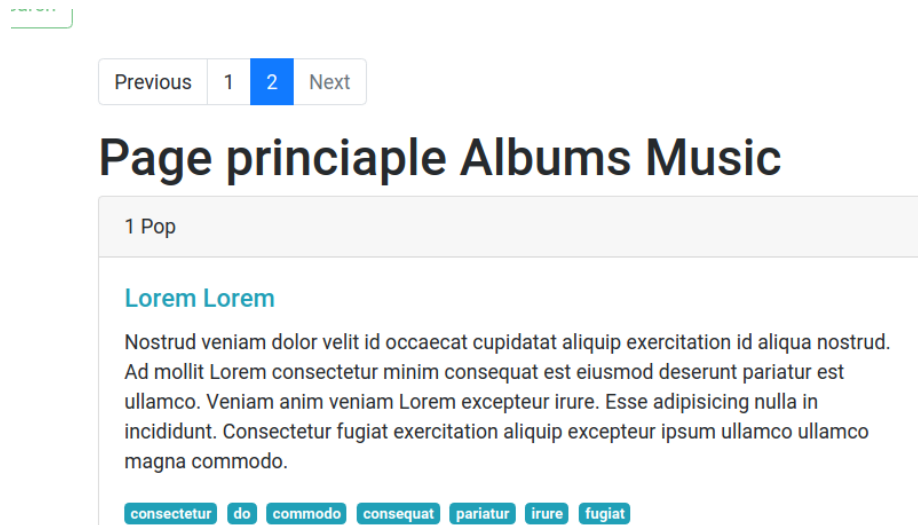


Figure 1: paginate

Fonctionnement : des boutons Previous et Next permettent respectivement de reculer et d'avancer dans l'affichage des albums.

Les boutons de pagination sont cliquables et permettent d'afficher un nombre d'albums par page prédéfini par vos soins (dans le code).

Créez une méthode paginate dans votre service, elle retournera un tableau d'albums slicés (utilisez la méthode slice de JS).

Dans le component `PaginateComponent` créez les méthodes de pagination classiques :

`next()` ; `previous()` et des propriétés classiques comme `perPage` (nombre d'album(s) par page), `total` (nombre total d'album(s)), ...

Le component enfant `PaginateComponent` mettra à jour le nombre d'albums à afficher sur la page « albums ».

Subscription

Dans la pagination il faut synchroniser les deux components pour que la page sélectionnée soit répercutée dans tous les components de pagination.

Dans le service `AlbumService` vous devez tout d'abord créer un `Subject`, c'est un `Observable` et `Subject` à la fois capable d'écouter et de réagir/envoyer une action en même temps.

Importez le service `RxJS` :

```
import { Subject } from 'rxjs';
```

Puis définissez dans le service un `Subject` :

```
// ...
export class AlbumService {

  private _albums: Album[] = ALBUMS;
  private _albumList: List[] = ALBUM_LISTS;

  // subject pour la pagination informer les autres components
  sendCurrentNumberPage = new Subject<number>();

  // ...
}
```

Puis implémentez la méthode dans le service comme suit, par exemple :

```
// méthode dans le service
currentPage(page: number) {
  return this.sendCurrentNumberPage.next(page);
}
```

Dans le component `PaginateComponent` vous devez utiliser le code qui suit il permet de récupérer l'information émise par le sujet :

```
this.aS.sendCurrentNumberPage.subscribe(numberPage => {
  // number page TODO
});
```

Hierarchie de l'application

Voici à quoi ressemble pour l'instant notre application :

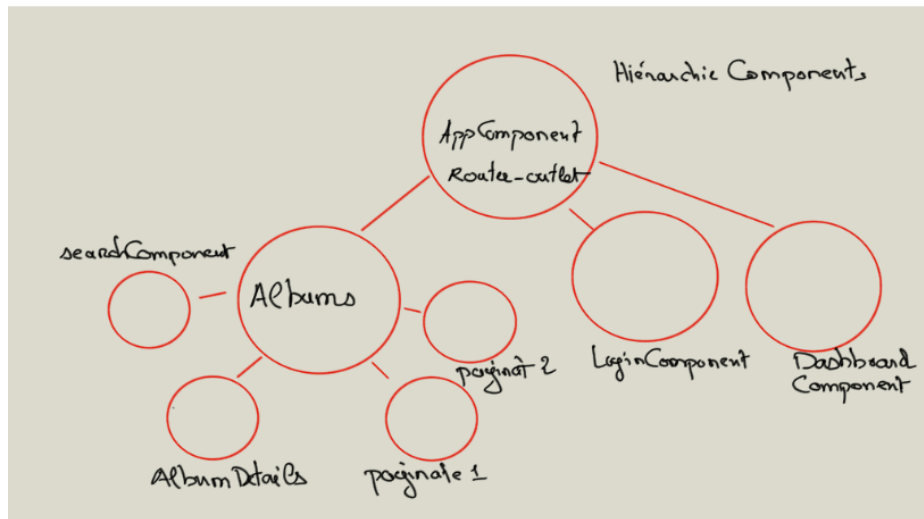


Figure 2: paginate