

Documentation de l'Application : Affichage Dynamique des Images et des Textes

Thème : Affichage dynamique et interactif d'images et de textes associant des processus marins et humains.

1. Introduction

Cette application Angular est conçue pour illustrer les correspondances entre des éléments marins et des parties du corps humain. L'application permet d'afficher des images associées à deux catégories principales : 'ocean' et 'humain', accompagnées de textes explicatifs. Elle propose une navigation dynamique et interactive, avec des interactions telles que l'affichage d'une boîte modale dédiée aux images humaines associées.

Le système est conçu pour être facilement extensible : ajouter de nouvelles paires d'images et de textes ne nécessite que quelques étapes simples, sans modification majeure du code.

2. Fonctionnement de l'Application

2.1. Structure Générale

- Composants :

- <app-image-text> : Affiche une image et son texte explicatif.
- <app-home> : Affiche les galeries interactives avec les images 'ocean' et permet d'ouvrir les images associées 'humain'.

- Affichage des données :

- Les images sont récupérées dynamiquement des dossiers `assets/ocean/` et `assets/humain/`.
- Chaque image est associée à un texte descriptif provenant de tableaux prédéfinis (un pour chaque type).

2.2. Fonctionnalités Clés

1. Affichage des Images :

- Les images de type 'ocean' et 'humain' sont affichées côte à côte.
- Chaque image est cliquable pour afficher une boîte d'information spécifique.

2. Association des Textes :

- Chaque image a un texte descriptif récupéré automatiquement en fonction de son type et de son index.

3. Interaction Utilisateur :

- L'utilisateur peut cliquer sur une image 'ocean' pour ouvrir une boîte dynamique affichant l'image 'humain' correspondante.

4. Responsive Design :

- L'interface est adaptée pour être utilisable sur des appareils de tailles différentes (mobile, tablette, bureau).

3. Ajouter un Nouveau Processus

L'ajout d'un nouveau processus (paire image/texte) est simple et rapide.

Étape 1 : Ajouter les Images

1. Image 'ocean' :

- Ajoutez une image dans le dossier ``assets/ocean/``.
- Nommez l'image avec un numéro suivant l'ordre séquentiel (par exemple : ``6.jpg`` si les fichiers existants vont jusqu'à ``5.jpg``).

2. Image 'humain' :

- Ajoutez l'image correspondante dans le dossier ``assets/humain/``.
- Nommez-la avec le même numéro que l'image 'ocean' (par exemple : ``6.jpg``).

Étape 2 : Ajouter le Texte Associé

Dans le fichier ``image-text.component.ts``, ajoutez les textes explicatifs pour les deux types dans les tableaux ``oceanText`` et ``humainText`` :

Exemple :

```
oceanText.push('Les mangroves filtrent les polluants et protègent le littoral.');
```

```
humainText.push('La peau forme une barrière protectrice, comme les mangroves pour la
```

```
côte.');
```

Étape 3 : Lancer ou Actualiser l'Application

1. Relancez le serveur Angular avec la commande : ``ng serve``.
2. Rechargez la page dans le navigateur. Les nouvelles images et textes seront affichés automatiquement.

4. Détails Techniques

4.1. Composant ``app-image-text``

Le composant `<app-image-text>` est responsable de l'affichage des images et des textes associés.

Entrées du Composant :

- ``imageUrl`` : Numéro de l'image (1, 2, ...).
- ``type`` : Type d'image ('ocean' ou 'humain').

Méthodes Clés :

- ``getImagePath()`` : Construit dynamiquement le chemin vers l'image.
- ``getText()`` : Retourne le texte associé à l'image et au type.

Exemple de Rendu HTML :

```
<div class='image-text-box'>
```

```
<img [src]='getImagePath()' alt='Image' class='img-fluid'>
<p>{{ getText() }}</p>
</div>
```

4.2. Composant `app-home`

Le composant <app-home> gère l'affichage des galeries d'images et l'ouverture des boîtes interactives.

Méthodes Clés :

- `displayHumanBox(index: number)` : Ouvre la boîte modale pour l'image 'humain' correspondant à l'index cliqué.
- `closeHumanBox()` : Ferme la boîte modale.

Exemple de Code HTML pour la Galerie :

```
<div class='row'>
  <div class='column' *ngFor='let index of [1, 2, 3, 4, 5]'>
    <app-image-text [imageUrl]='index' [type]='ocean'
(click)='displayHumanBox(index)'></app-image-text>
  </div>
</div>
```

5. Extensions et Améliorations

- **Support de Nouveaux Types :** Ajouter d'autres types d'images (ex. 'montagne', 'forêt') en créant de nouveaux dossiers dans `assets/` et en mettant à jour le composant.
- **Base de Données Dynamique :** Intégrer un backend (ex. Firebase, Node.js) pour charger les images et textes dynamiquement au lieu de les définir statiquement.
- **Amélioration Visuelle :** Ajouter des animations CSS ou des transitions pour rendre l'expérience utilisateur plus fluide.