

Rapport

Implémentation d'un drawer

Ce document présente un aperçu de l'implémentation d'un drawer développé avec React, TypeScript, en utilisant les bibliothèques Ant Design, Ant Design Pro Layout, et JsonForms. L'objectif est de remplacer les onglets traditionnels par un drawer dynamique permettant d'afficher différents formulaires en cliquant.

L'interface est responsive : elle s'adapte aux écrans d'ordinateur comme aux appareils mobiles.

Technologies Utilisées

- **React** : Construction de l'interface utilisateur de manière déclarative et modulaire.
- **TypeScript** : Typage statique pour renforcer la robustesse du code.
- **Ant Design** : Composants UI pour une interface professionnelle et responsive.
- **Ant Design Pro Layout** : Gestion de la structure de l'interface avec un header et un menu latéral.
- **JsonForms** : Génération dynamique de formulaires à partir de schémas JSON.

Organisation du code

Composants Principaux

Le projet est organisé autour de plusieurs composants réutilisables, chacun ayant une responsabilité claire :

Categorization : Afficher une liste de catégories pour un formulaire donné. Ce composant est utilisé dans le tiroir (Drawer) pour afficher les catégories disponibles.

title : Le titre de la catégorie.

categories : La liste des catégories.

activeCategoryId : L'index de la catégorie active.

onCategorySelect : Fonction de rappel pour gérer la sélection d'une catégorie.

FormRenderer : Afficher un formulaire dynamique en utilisant JsonForms. Ce composant est utilisé pour afficher le formulaire correspondant à la catégorie sélectionnée.

schema : Le schéma JSON du formulaire.

uiSchema : Le schéma UI pour le rendu du formulaire.

data : Les données du formulaire.

onChange : Fonction de rappel pour gérer les changements de données.

HomePage : Afficher une page d'introduction avec un message de bienvenue et un bouton pour commencer. Ce composant est affiché lorsque l'utilisateur arrive sur la page d'accueil.

onGetStarted : Fonction de rappel pour gérer le clic sur le bouton "Get Started".

DrawerContent : Afficher le contenu du Drawer, y compris les catégories et un bouton pour revenir à l'accueil et un autre bouton pour fermer le drawer.

formSchemas : La liste des schémas de formulaire.

activeFormIndex : L'index du formulaire actif.

activeCategoryIndexes : Les index des catégories actives pour chaque formulaire.

onCategorySelect : Fonction de rappel pour gérer la sélection d'une catégorie.

onHomeClick : Fonction de rappel pour revenir à la page d'accueil.

App : Le composant principal qui orchestre l'affichage des autres composants et gère l'état global de l'application.

Organisation des Fichiers

Le projet est organisé de la manière suivante :

- **components/** : Contient tous les composants réutilisables.
- **schemas/** : Contient les schémas JSON et UI pour les formulaires.
- **App.tsx** : Le composant principal de l'application.
- **index.tsx** : Le point d'entrée de l'application.

Conclusion

Ce projet démontre l'intégration fluide de plusieurs bibliothèques modernes (React, TypeScript, Ant Design, JsonForms) pour créer une interface dynamique et responsive.

L'utilisation de schémas JSON permet une gestion flexible des formulaires, et l'adaptation conditionnelle garantit une expérience utilisateur optimale sur différents appareils.