

Documentation du début de projet en React Native

Documentation du composant AppNavigator

Introduction

Ce code est une implémentation initiale pour la navigation entre deux écrans, "Login" et "Register", dans une application React Native. Il utilise la bibliothèque @react-navigation/native pour gérer la navigation entre ces écrans.

Structure du code

Importations nécessaires:

- React : La bibliothèque principale pour construire des composants React.
- NavigationContainer: Un composant de conteneur qui gère et contient l'arbre de navigation.
- createStackNavigator: Une fonction qui permet de créer un stack navigator.
- Login et Register: Deux composants d'écran qui seront utilisés comme écrans de connexion et d'inscription.

Définition du Stack Navigator:

```
const Stack = createStackNavigator();
```

Cette ligne crée un nouveau Stack Navigator. Le Stack Navigator est un type de navigateur qui présente les écrans de manière empilée, où chaque nouvel écran est placé sur le sommet de la pile.

Composant AppNavigator:

Le composant AppNavigator est la définition principale de la logique de navigation. Il contient :

- Un NavigationContainer qui englobe tout l'arbre de navigation.
- Un Stack.Navigator qui définira les différentes routes ou écrans de l'application

Détails du Stack.Navigator

- `initialRouteName`: Spécifie l'écran qui sera affiché en premier lors de la première ouverture de l'application. Dans ce cas, il est défini sur "Login".
- `screenOptions`: Ce sont les options qui seront appliquées à tous les écrans à l'intérieur de ce "Navigator". Ici, l'option "`headerShown: false`" garantit que l'en-tête par défaut de l'écran ne sera pas affiché.
- Les écrans disponibles sont définis avec le composant "Stack.Screen". Chaque écran a un `name` qui est utilisé pour référencer cet écran et un `component` qui est le composant effectivement rendu lorsque cet écran est actif.

Comment l'utiliser ?

Pour intégrer ce Navigator dans une application, vous devez l'importer et le rendre dans votre composant principal.

```
import AppNavigator from './path-to-AppNavigator-file';
```

Puis, dans le composant de rendu :

```
function App() {  
  return <AppNavigator />;  
}
```

Conclusion

Ce composant est une base pour une application avec une navigation entre un écran de connexion et un écran d'inscription. Grâce à la bibliothèque de navigation, l'extension de cette base pour inclure d'autres écrans et fonctionnalités est facilement réalisable.

Documentation du composant App

Introduction

Le composant App est une page de connexion. Il offre un moyen pour un utilisateur de saisir son nom d'utilisateur et son mot de passe, de se connecter au backend et d'être redirigé vers l'inscription s'il n'a pas encore de compte.

Description détaillée

- État local :
 - username : Stocke la valeur saisie dans le champ du nom d'utilisateur.
 - password : Stocke la valeur saisie dans le champ du mot de passe.
- Gestion de la connexion :
 - La fonction "handleLogin" est responsable de faire une requête POST vers un endpoint backend pour la connexion. La réponse est analysée et si un token est renvoyé, cela indique une connexion réussie. Sinon, un message d'erreur est affichée.
- Rendu JSX :
 - En-tête: Affiche le titre "Log in" et une description.
 - Image: Affiche le logo provenant de logo.png.
 - Champs de saisie: Deux champs pour saisir le nom d'utilisateur et le mot de passe.
 - Bouton de connexion: Un bouton qui déclenche la fonction "handleLogin" lorsqu'il est pressé.
 - Redirection vers l'inscription: Un texte cliquable qui redirige l'utilisateur vers une page d'inscription via la navigation.
- Styles :
 - Un ensemble de styles est défini à l'aide de "StyleSheet.create" pour donner à l'application un aspect visuel agréable et cohérent.

Points importants

- Sécurité: Le mot de passe est caché lors de la saisie grâce à "secureTextEntry={true}".
- Gestion des erreurs: En cas d'échec de la connexion ou d'erreur réseau, une alerte est affichée à l'utilisateur.
- Redirection vers la page d'inscription: Si l'utilisateur n'a pas de compte, il peut être redirigé vers une page d'inscription en cliquant sur "Register".

Conclusion

Ce composant est une page de connexion dans une application React Native. Il intègre des champs de saisie, des boutons et une interaction avec un service backend, tout en offrant un aspect visuel attrayant grâce aux styles.

Documentation du composant Register

Introduction

Le composant Register est une page qui permet aux utilisateurs de s'inscrire en fournissant leur email, nom, prénom et mot de passe. Lorsque l'utilisateur clique sur le bouton "Register", les détails sont envoyés à un backend pour traitement.

Description détaillée

- État local:
 - email: Stocke l'adresse e-mail entrée par l'utilisateur.
 - name: Stocke le prénom entré par l'utilisateur.
 - surname: Stocke le nom de famille entré par l'utilisateur.
 - password: Stocke le mot de passe entré par l'utilisateur.
- Fonction "handleRegister":
 - Cette fonction s'exécute lorsque l'utilisateur clique sur le bouton "Register".
 - Elle envoie une requête POST au backend pour créer un nouvel utilisateur.
 - En cas de succès, une alerte est affichée et l'utilisateur est redirigé vers l'écran de connexion.
 - En cas d'échec, une alerte est affichée avec le message d'erreur.
- Rendu JSX:
 - Header: Affiche le titre "Register" et une description.
 - Formulaires: Permet à l'utilisateur d'entrer son email, son prénom, son nom de famille et son mot de passe.
 - Bouton Register: Lorsqu'il est cliqué, la fonction handleRegister est appelée.
 - Lien de connexion: Si l'utilisateur a déjà un compte, il peut cliquer sur ce lien pour aller à l'écran de connexion.
- Styles:

- Les styles sont définis pour rendre le composant esthétiquement plaisant. Il y a des styles pour le conteneur, le header, les champs de formulaire, le bouton d'inscription, et le lien vers l'écran de connexion.

Fonctionnalités principales

- Enregistrement: L'utilisateur peut s'inscrire en fournissant son email, nom, prénom et mot de passe.
- Feedback de l'API: En fonction de la réponse de l'API, l'utilisateur reçoit un feedback sous forme d'une alerte.
- Navigation: L'utilisateur peut naviguer vers l'écran de connexion s'il possède déjà un compte.

Conclusion

Le composant Register fournit une fonctionnalité essentielle pour permettre aux nouveaux utilisateurs de s'inscrire à l'application.