**Deuxième Partie : Consommation API et Push Notification**

1. **Consommation API**

On doit ajouter pour l’admin un button dans **HomeScreen** pour ajouter un provider

import React, { useState,useEffect } from 'react'

import {

StyleSheet,

Text,

View,

Image,

FlatList,TouchableOpacity

} from 'react-native'

import axios from "axios";

import asyncStorage from "@react-native-async-storage/async-storage/src/AsyncStorage";

import Icon from 'react-native-vector-icons/FontAwesome';

const HomeScreen = () => {

const [providers, setProviders] = useState([]);

const fetchProviders = async () => {

const u= await asyncStorage.getItem("token");

axios.defaults.headers['Authorization'] = 'Bearer ' + u;

const a = await asyncStorage.getItem("id");

const res = await axios

.get("https://ams.smart-it-partner.com/api/providers")

.then(response=>response.data["hydra:member"])

setProviders(res);

console.log(res)

}

useEffect(() => {

fetchProviders();

}, []);

const addProvider = () => {

console.log('Ajouter un provider');

};

return (

<View style={{ flex: 1 }}>

<TouchableOpacity style={styles.addButton} onPress={addProvider}>

<Icon name="plus" size={20} color="white" />

<Text style={styles.addButtonText}>Add Provider</Text>

</TouchableOpacity>

<FlatList

enableEmptySections={true}

data={providers}

keyExtractor={item => item.id}

renderItem={({ item }) => {

return (

<View style={styles.box}>

<Image style={styles.image} source={{ uri:'https://ams.smart-it-partner.com/uploads/provider/'+ item.photo }} />

<View style={styles.boxContent}>

<Text style={styles.title}>{item.name}</Text>

<Text style={styles.description}>{item.adress}</Text>

<Text style={styles.description}>{item.email}</Text>

</View>

</View>

)

}}

/></View>

)

}

const styles = StyleSheet.create({

image: {

width: 100,

height: 100,

},

box: {

padding: 20,

marginTop: 5,

marginBottom: 5,

backgroundColor: 'white',

flexDirection: 'row',

},

boxContent: {

flex: 1,

flexDirection: 'column',

alignItems: 'flex-start',

marginLeft: 10,

},

title: {

fontSize: 18,

color: '#151515',

},

description: {

fontSize: 15,

color: '#646464',

},

view: {

backgroundColor: '#eee',

},

addButton: {

backgroundColor: '#07B0A8,

paddingHorizontal: 20,

paddingVertical: 10,

flexDirection: 'row',

alignItems: 'center',

justifyContent: 'center',

borderRadius: 5,

marginTop: 10, // Marge en haut

marginLeft: 200, // Marge à gauche

marginRight:10

},

addButtonText: {

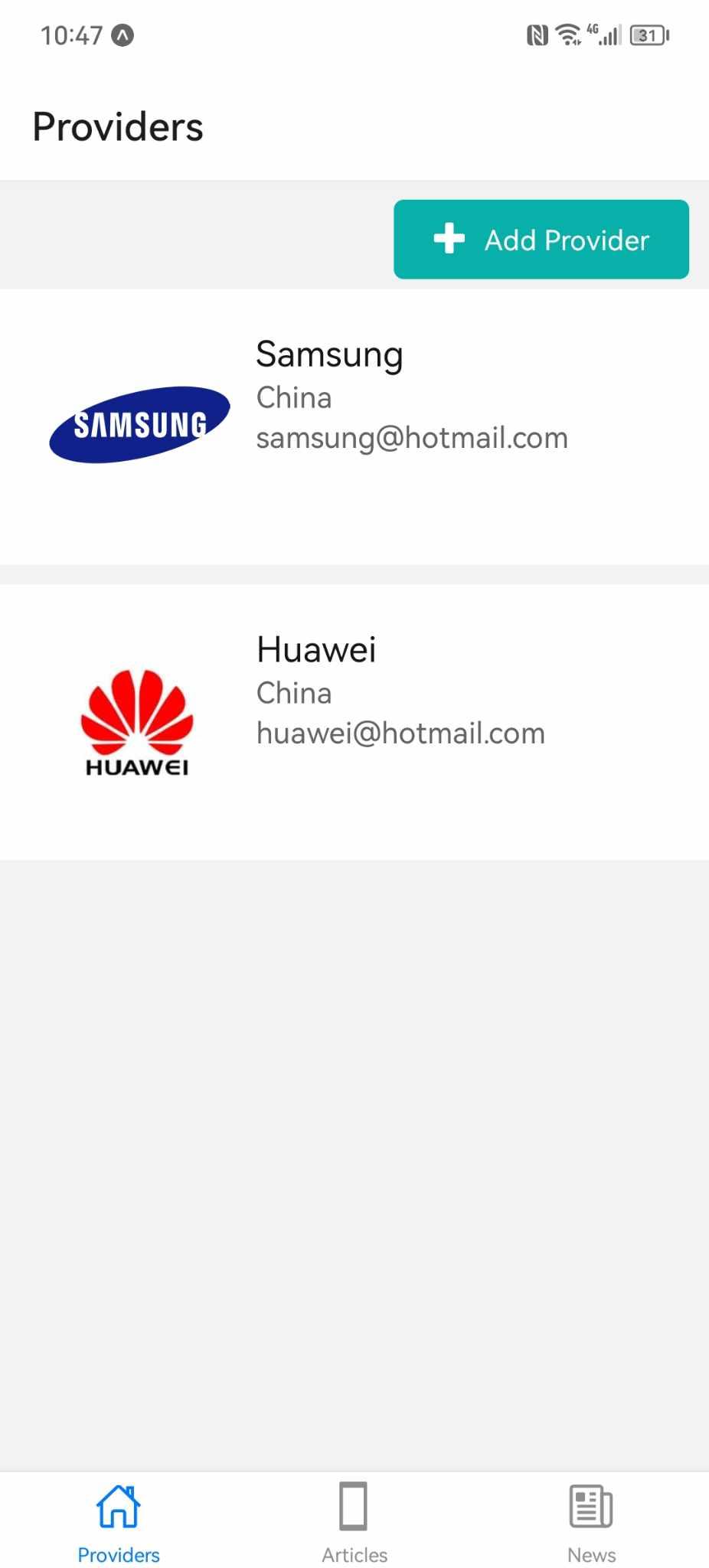
color: 'white',

marginLeft: 10,

},

})

export default HomeScreen;



On doit ajouter une nouvelle screen **AddProviderScreen.js** pour ajouter un provider

import React, { useState,useEffect } from 'react'

import {

StyleSheet,

Text,

View,

Image,

FlatList,TouchableOpacity

} from 'react-native'

const addProviderScreen = () => {

return (

<View style={{ flex: 1 }}>

</View>

)

}

const styles = StyleSheet.create({

})

export default addProviderScreen;

On doit modifier **TabNavigator**

**import { StatusBar } from 'expo-status-bar';**

**import { StyleSheet } from 'react-native'**

**import {createBottomTabNavigator } from '@react-navigation/bottom-tabs';**

**import HomeScreen from '../screens/HomeScreen';**

**import Ionicons from 'react-native-vector-icons/Ionicons';**

**import NewScreen from '../screens/NewScreen';**

**import ArticleScreen from '../screens/ArticleScreen';**

**import addProviderScreen from '../screens/AddProviderScreen';**

**import { createNativeStackNavigator } from '@react-navigation/native-stack';**

**const tab = createBottomTabNavigator();**

**const Stack = createNativeStackNavigator();**

**const HomeStack = () => (**

**<Stack.Navigator>**

**<Stack.Screen name="Home" component={HomeScreen} options={{headerShown: false}} />**

**<Stack.Screen**

**name="Add Provider"**

**component={addProviderScreen}**

**options={({route}) => ({**

**title: route.params?.title,**

**headerStyle: {**

**backgroundColor: '#DFF8F7',**

**}**

**})}**

**/>**

**</Stack.Navigator>**

**);**

**export default function TabNavigator() {**

**return (**

**<tab.Navigator**

**screenOptions={({ route }) => ({**

**tabBarIcon: ({ focused, color, size }) => {**

**let iconName;**

**if (route.name == "Home") { iconName = "home-outline"; }**

**else if (route.name == "Articles") { iconName = "phone-portrait-sharp"; }**

**else if (route.name == "News") { iconName = "newspaper-outline"; }**

**return (**

**<Ionicons**

**name={iconName}**

**color={color}**

**size={size}**

**/>**

**);**

**},**

**})}>**

**<tab.Screen name="Home" component={HomeStack} options={{ title: 'Providers' }}/>**

**<tab.Screen name="Articles" component={ArticleScreen} options={{ title: 'Articles' }}/>**

**<tab.Screen name="News" component={NewScreen} options={{ title: 'News' }}/>**

**</tab.Navigator>**

**);**

**}**

**const styles = StyleSheet.create({**

**container: {**

**flex: 1,**

**backgroundColor: '#fff',**

**alignItems: 'center',**

**justifyContent: 'center',**

**},**

**});**

**AddProviderScreen**

**import React, { useState } from 'react';**

**import { View, Text, Image, TextInput, StyleSheet, TouchableOpacity } from 'react-native';**

**import axios from "axios";**

**import { useNavigation } from '@react-navigation/native'**

**const AddProviderScreen = () => {**

**const [name, setName] = useState();**

**const [adress, setAdress] = useState();**

**const [email, setEmail] = useState();**

**const navigation=useNavigation();**

**const addProvider = async () => {**

**const provider = {**

**name: name,**

**adress: adress,**

**email: email,**

**};**

**try {**

**const response = await axios.post("https://ams.smart-it-partner.com/api/providers", provider, {**

**headers: {'Content-Type': 'application/ld+json'}**

**});**

**navigation.navigate('Home');**

**} catch (error) {**

**// Gérez l'erreur ici**

**console.error(error);**

**}**

**};**

**return (**

**<View style={styles.container}>**

**<View style={styles.form}>**

**<Text style={styles.label}>Name</Text>**

**<TextInput**

**style={styles.input}**

**placeholder="Enter Name"**

**value={name}**

**onChangeText={setName}**

**/>**

**<Text style={styles.label}>Email</Text>**

**<TextInput**

**style={styles.input}**

**placeholder="Enter Email"**

**value={email}**

**onChangeText={setEmail}**

**/>**

**<Text style={styles.label}>Adress</Text>**

**<TextInput**

**style={styles.input}**

**placeholder="Enter Adress"**

**value={adress}**

**onChangeText={setAdress}**

**/>**

**<TouchableOpacity style={styles.button} onPress={() => addProvider()}>**

**<Text style={styles.buttonText}>Valider</Text>**

**</TouchableOpacity>**

**</View>**

**</View>**

**);**

**};**

**const styles = StyleSheet.create({**

**container: {**

**flex: 1,**

**alignItems: 'center',**

**backgroundColor:'#fff',**

**},**

**form: {**

**width: '80%',**

**},**

**label: {**

**marginTop: 20,**

**marginBottom:5**

**},**

**input: {**

**borderColor: '#ccc',**

**borderWidth: 1,**

**borderRadius: 5,**

**padding: 10,**

**fontSize: 18,**

**},**

**button: {**

**marginTop: 20,**

**backgroundColor: '#07B0A8',**

**borderRadius: 5,**

**paddingVertical: 10,**

**paddingHorizontal: 20,**

**},**

**buttonText: {**

**color: '#fff',**

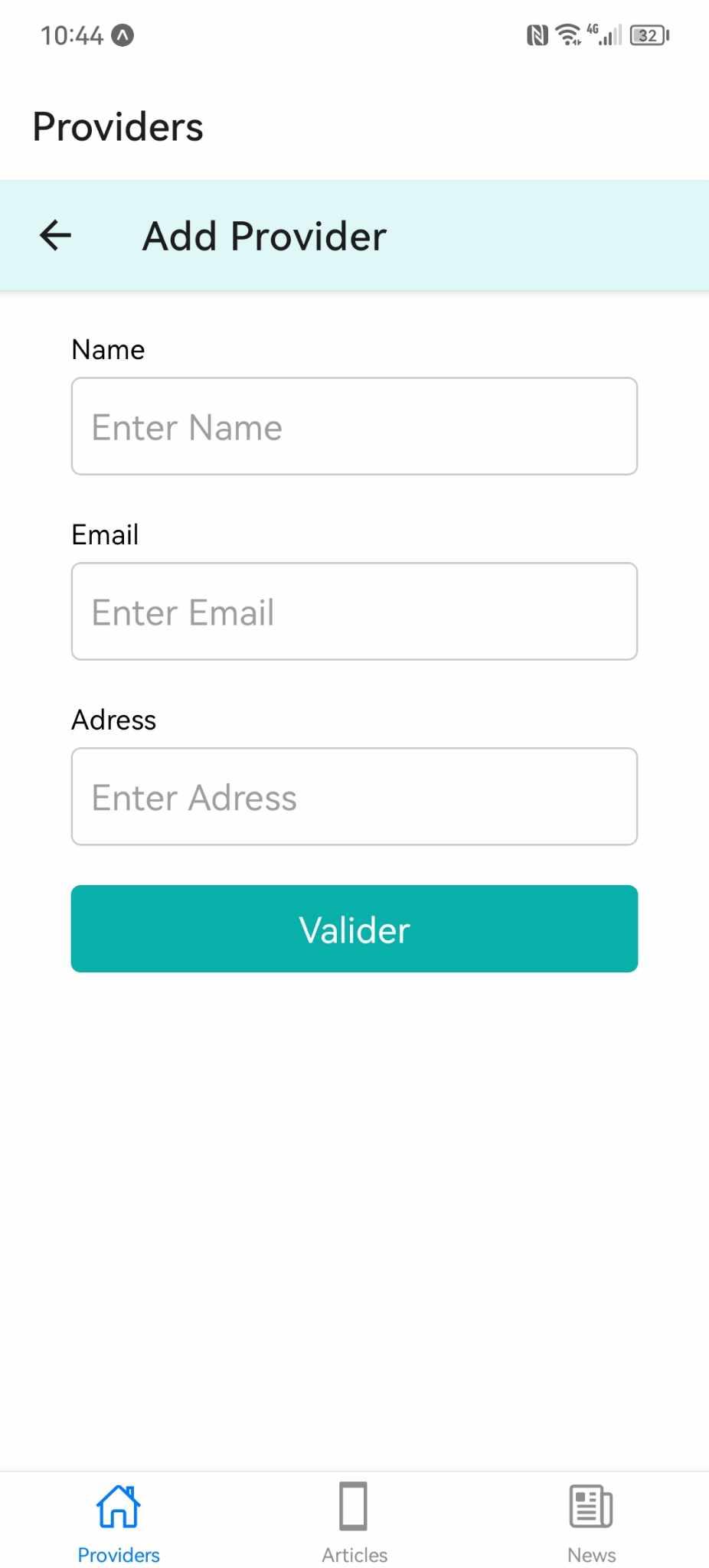
**fontSize: 18,**

**textAlign:'center',**

**},**

**});**

**export default AddProviderScreen;**

****

**Modifier dans le home Screen**

const addProvider = () => { navigation.navigate('Add Provider'); };

import { useNavigation } from '@react-navigation/native'

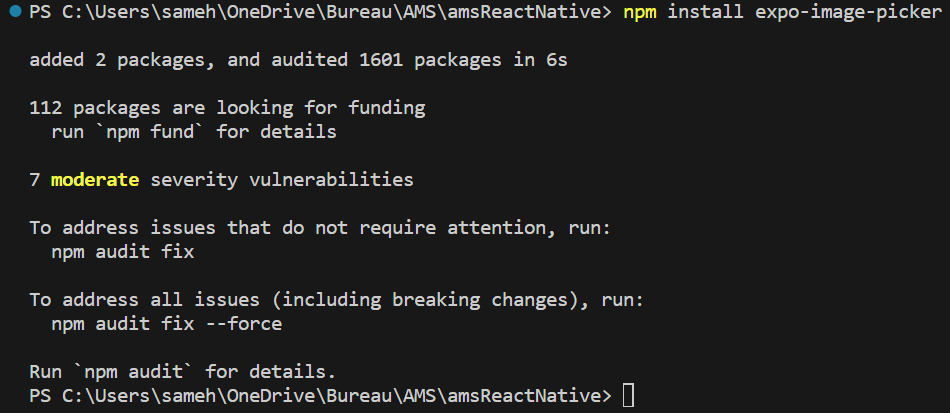
const navigation=useNavigation();

**Pour forcer le refresh**

navigation.reset({ index: 0, routes: [{ name: 'Home', params: { refresh: true } }], });

**Upload Image**

On doit installer npm install expo-image-picker



On doit modifier **AddProviderScreen**

import React, { useState } from 'react';

import { View, Text, Image, TextInput, StyleSheet, TouchableOpacity } from 'react-native';

import axios from "axios";

import { useNavigation } from '@react-navigation/native';

import \* as ImagePicker from 'expo-image-picker';

import Ionicons from "react-native-vector-icons/Ionicons";

const AddProviderScreen = () => {

const [name, setName] = useState();

const [adress, setAdress] = useState();

const [email, setEmail] = useState();

const navigation=useNavigation();

const [pickedImagePath, setPickedImagePath] = useState('');

const showImagePicker = async () => {

// Ask the user for the permission to access the media library

const permissionResult = await ImagePicker.requestMediaLibraryPermissionsAsync();

if (permissionResult.granted === false) {

alert("You've refused to allow this appp to access your photos!");

return;

}

const result = await ImagePicker.launchImageLibraryAsync();

if (!result.didCancel) {

setPickedImagePath(result.uri);

} else {

console.log('Sélection d\'image annulée');

}

}

// This function is triggered when the "Open camera" button pressed

const openCamera = async () => {

// Ask the user for the permission to access the camera

const permissionResult = await ImagePicker.requestCameraPermissionsAsync();

if (permissionResult.granted === false) {

alert("You've refused to allow this appp to access your camera!");

return;

}

const result = await ImagePicker.launchCameraAsync();

if (!result.didCancel) {

setPickedImagePath(result.uri);

}

}

const addProvider = async () => {

const provider = {

name: name,

adress: adress,

email: email,

};

try {

const response = await axios.post("https://ams.smart-it-partner.com/api/providers", provider, {

headers: {'Content-Type': 'application/ld+json'}

});

navigation.navigate('Home');

} catch (error) {

// Gérez l'erreur ici

console.error(error);

}

};

return (

<View style={styles.container}>

<View style={styles.imageContainer}>

{pickedImagePath !== '' && <Image

source={{ uri: pickedImagePath }}

style={styles.image}

/>

}

</View>

<View style={styles.form}>

<View style={styles.buttonContainer}>

<Text style={styles.label}>Image</Text>

<TouchableOpacity style={styles.uploadButton} onPress={showImagePicker}>

<Ionicons

name="image"

size={30}

color="#149750"

/>

</TouchableOpacity>

<TouchableOpacity style={styles.cameraButton} onPress={openCamera}>

<Ionicons

name="camera"

size={30}

color="#165FAD"

/>

</TouchableOpacity>

</View>

<Text style={styles.label}>Name</Text>

<TextInput

style={styles.input}

placeholder="Enter Name"

value={name}

onChangeText={setName}

/>

<Text style={styles.label}>Email</Text>

<TextInput

style={styles.input}

placeholder="Enter Email"

value={email}

onChangeText={setEmail}

/>

<Text style={styles.label}>Adress</Text>

<TextInput

style={styles.input}

placeholder="Enter Adress"

value={adress}

onChangeText={setAdress}

/>

<TouchableOpacity style={styles.button} onPress={() => addProvider()}>

<Text style={styles.buttonText}>Valider</Text>

</TouchableOpacity>

</View>

</View>

);

};

const styles = StyleSheet.create({

container: {

flex: 1,

alignItems: 'center',

backgroundColor:'#fff',

},

form: {

width: '80%',

},

label: {

marginTop: 20,

marginBottom:5

},

input: {

borderColor: '#ccc',

borderWidth: 1,

borderRadius: 5,

padding: 10,

fontSize: 18,

},

button: {

marginTop: 20,

backgroundColor: '#07B0A8',

borderRadius: 5,

paddingVertical: 10,

paddingHorizontal: 20,

},

buttonText: {

color: '#fff',

fontSize: 18,

textAlign:'center',

},

uploadButton: {

backgroundColor: 'white',

padding: 10,

marginLeft:0,

borderRadius: 15,

backgroundColor:'#e7e7e7',

},

cameraButton: {

backgroundColor: 'white',

padding: 10,

marginRight:50,

borderRadius: 15,

backgroundColor:'#e7e7e7',

},

buttonContainer: {

flexDirection: 'row',

justifyContent: 'space-between',

marginVertical: 5, marginTop: 10,

},

imageContainer: {

alignItems: 'center',

marginLeft:10,

marginVertical: 10,

width:'80%'

},

image: {

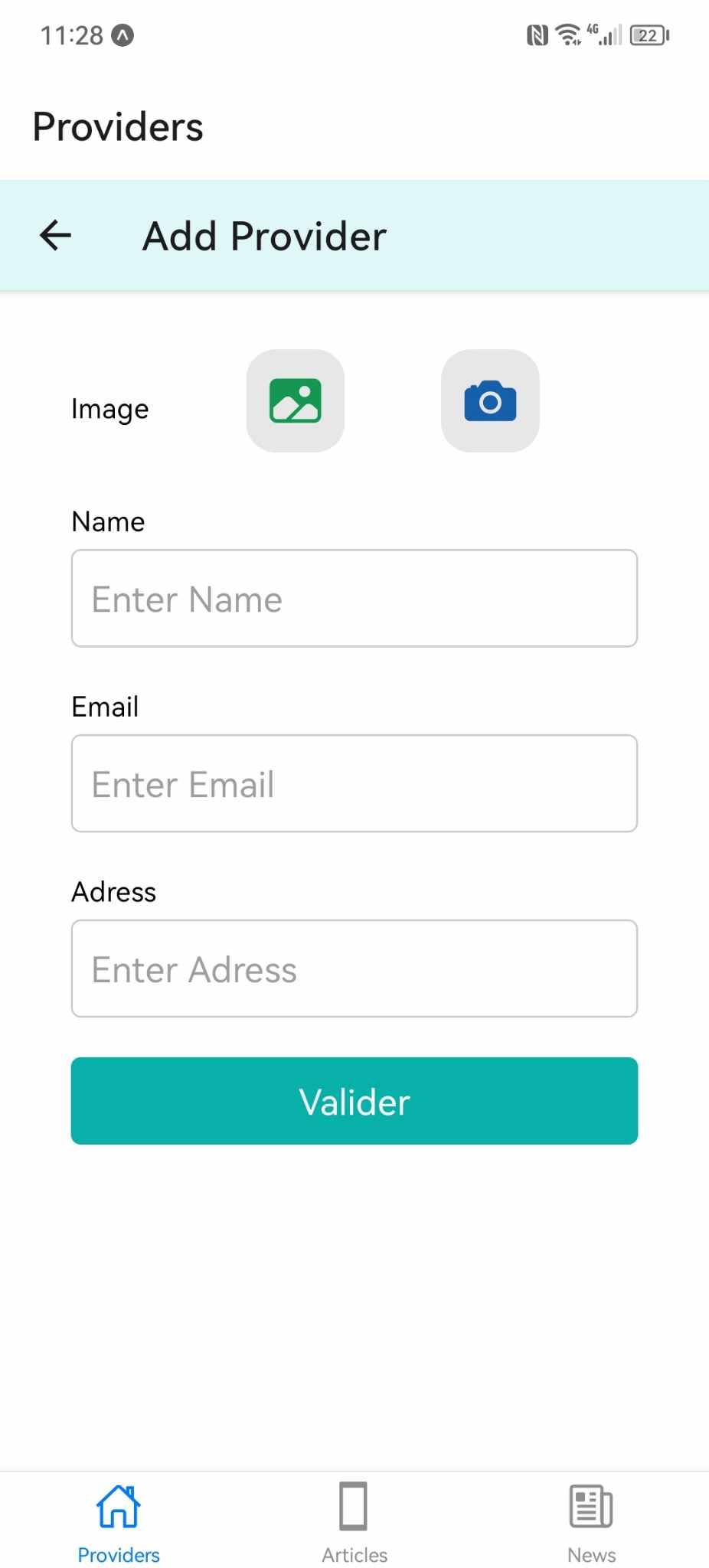
width: '100%',

height: 250,

},

});

export default AddProviderScreen;



On doit maintenant modifier la fonction **addProvider**

**import React, { useState } from 'react';**

**import { View, Text, Image, TextInput, StyleSheet, TouchableOpacity,ScrollView } from 'react-native';**

**import axios from "axios";**

**import { useNavigation } from '@react-navigation/native';**

**import \* as ImagePicker from 'expo-image-picker';**

**import Ionicons from "react-native-vector-icons/Ionicons";**

**import asyncStorage from "@react-native-async-storage/async-storage/src/AsyncStorage";**

**const AddProviderScreen = () => {**

**const [name, setName] = useState();**

**const [adress, setAdress] = useState();**

**const [email, setEmail] = useState();**

**const navigation=useNavigation();**

**const [pickedImagePath, setPickedImagePath] = useState('');**

**const showImagePicker = async () => {**

**// Ask the user for the permission to access the media library**

**const permissionResult = await ImagePicker.requestMediaLibraryPermissionsAsync();**

**if (permissionResult.granted === false) {**

**alert("You've refused to allow this appp to access your photos!");**

**return;**

**}**

**const result = await ImagePicker.launchImageLibraryAsync();**

**if (!result.didCancel) {**

**setPickedImagePath(result.uri);**

**} else {**

**console.log('Sélection d\'image annulée');**

**}**

**}**

**// This function is triggered when the "Open camera" button pressed**

**const openCamera = async () => {**

**// Ask the user for the permission to access the camera**

**const permissionResult = await ImagePicker.requestCameraPermissionsAsync();**

**if (permissionResult.granted === false) {**

**alert("You've refused to allow this appp to access your camera!");**

**return;**

**}**

**const result = await ImagePicker.launchCameraAsync();**

**if (!result.didCancel) {**

**setPickedImagePath(result.uri);**

**}**

**}**

**const selectedImage = pickedImagePath;**

**const addProvider = async () => {**

**let localUri = selectedImage;**

**const a = await asyncStorage.getItem("id");**

**let filename = localUri ? localUri.split('/').pop() : '';**

**let match = /\.(\w+)$/.exec(filename);**

**let imageType = match ? `image/${match[1]}` : `image`;**

**const formData = new FormData();**

**if (localUri) {**

**formData.append('image', { uri: localUri, name: filename, type: imageType });**

**}**

**formData.append('adress', adress);**

**formData.append('name', name);**

**formData.append('email', email);**

**try {**

**const response = await axios.post("https://ams.smart-it-partner.com/addprovider/mobile", formData, {**

**headers: { 'Content-Type': 'multipart/form-data' },**

**});**

**navigation.reset({**

**index: 0,**

**routes: [{ name: 'Home', params: { refresh: true } }],**

**});**

**} catch (error) {**

**Alert.alert('Erreur', 'Une erreur est survenue lors de l\'ajout du publication', error);**

**}**

**};**

**return (**

**<View style={styles.container}>**

**<View style={styles.imageContainer}>**

**{pickedImagePath !== '' && <Image**

**source={{ uri: pickedImagePath }}**

**style={styles.image}**

**/>**

**}**

**</View>**

**<ScrollView style={styles.form}>**

**<View style={styles.buttonContainer}>**

**<Text style={styles.label}>Image</Text>**

**<TouchableOpacity style={styles.uploadButton} onPress={showImagePicker}>**

**<Ionicons**

**name="image"**

**size={30}**

**color="#149750"**

**/>**

**</TouchableOpacity>**

**<TouchableOpacity style={styles.cameraButton} onPress={openCamera}>**

**<Ionicons**

**name="camera"**

**size={30}**

**color="#165FAD"**

**/>**

**</TouchableOpacity>**

**</View>**

**<Text style={styles.label}>Name</Text>**

**<TextInput**

**style={styles.input}**

**placeholder="Enter Name"**

**value={name}**

**onChangeText={setName}**

**/>**

**<Text style={styles.label}>Email</Text>**

**<TextInput**

**style={styles.input}**

**placeholder="Enter Email"**

**value={email}**

**onChangeText={setEmail}**

**/>**

**<Text style={styles.label}>Adress</Text>**

**<TextInput**

**style={styles.input}**

**placeholder="Enter Adress"**

**value={adress}**

**onChangeText={setAdress}**

**/>**

**<TouchableOpacity style={styles.button} onPress={() => addProvider()}>**

**<Text style={styles.buttonText}>Valider</Text>**

**</TouchableOpacity>**

**</ScrollView>**

**</View>**

**);**

**};**

**const styles = StyleSheet.create({**

**container: {**

**flex: 1,**

**alignItems: 'center',**

**backgroundColor:'#fff',**

**},**

**form: {**

**width: '80%',**

**},**

**label: {**

**marginTop: 20,**

**marginBottom:5**

**},**

**input: {**

**borderColor: '#ccc',**

**borderWidth: 1,**

**borderRadius: 5,**

**padding: 10,**

**fontSize: 18,**

**},**

**button: {**

**marginTop: 20,**

**backgroundColor: '#07B0A8',**

**borderRadius: 5,**

**paddingVertical: 10,**

**paddingHorizontal: 20,**

**},**

**buttonText: {**

**color: '#fff',**

**fontSize: 18,**

**textAlign:'center',**

**},**

**uploadButton: {**

**backgroundColor: 'white',**

**padding: 10,**

**marginLeft:0,**

**borderRadius: 15,**

**backgroundColor:'#e7e7e7',**

**},**

**cameraButton: {**

**backgroundColor: 'white',**

**padding: 10,**

**marginRight:50,**

**borderRadius: 15,**

**backgroundColor:'#e7e7e7',**

**},**

**buttonContainer: {**

**flexDirection: 'row',**

**justifyContent: 'space-between',**

**marginVertical: 5, marginTop: 10,**

**},**

**imageContainer: {**

**alignItems: 'center',**

**marginLeft:10,**

**marginVertical: 10,**

**width:'60%'**

**},**

**image: {**

**width: '100%',**

**height: 150,**

**},**

**});**

**export default AddProviderScreen;**

Ajouter lien pour Supprimer providerdans **HomeScreen**

**import React, { useState,useEffect } from 'react'**

**import {**

**StyleSheet,**

**Text,**

**View,**

**Image,**

**FlatList,TouchableOpacity**

**} from 'react-native'**

**import axios from "axios";**

**import asyncStorage from "@react-native-async-storage/async-storage/src/AsyncStorage";**

**import Icon from 'react-native-vector-icons/FontAwesome'; // Exemple d'utilisation de FontAwesome**

**import { useNavigation } from '@react-navigation/native'**

**const HomeScreen = () => {**

**const [providers, setProviders] = useState([]);**

**const navigation=useNavigation();**

**const fetchProviders = async () => {**

**const u= await asyncStorage.getItem("token");**

**axios.defaults.headers['Authorization'] = 'Bearer ' + u;**

**const a = await asyncStorage.getItem("id");**

**const res = await axios**

**.get("https://ams.smart-it-partner.com/api/providers")**

**.then(response=>response.data["hydra:member"])**

**setProviders(res);**

**console.log(res)**

**}**

**useEffect(() => {**

**fetchProviders();**

**}, []);**

**const addProvider = () => {**

**navigation.navigate('Add Provider');**

**};**

**const deleteProvider = async (idprovider) => {**

**const u= await asyncStorage.getItem("token");**

**axios.defaults.headers['Authorization'] = 'Bearer ' + u;**

**await axios.delete( "https://ams.smart-it-partner.com/api/providers/"+idprovider)**

**.then( response => {**

**response.data;**

**})**

**navigation.reset({**

**index: 0,**

**routes: [{ name: 'Home', params: { refresh: true } }],**

**});**

**}**

**return (**

**<View style={{ flex: 1 }}>**

**<TouchableOpacity style={styles.addButton} onPress={addProvider}>**

**<Icon name="plus" size={20} color="white" />**

**<Text style={styles.addButtonText}>Add Provider</Text>**

**</TouchableOpacity>**

**<FlatList**

**enableEmptySections={true}**

**data={providers}**

**keyExtractor={item => item.id}**

**renderItem={({ item }) => {**

**return (**

**<View style={styles.box}>**

**<Image style={styles.image} source={{ uri:'https://ams.smart-it-partner.com/uploads/provider/'+ item.photo }} />**

**<View style={styles.boxContent}>**

**<Text style={styles.title}>{item.name}</Text>**

**<Text style={styles.description}>{item.adress}</Text>**

**<Text style={styles.description}>{item.email}</Text>**

**<View style={styles.buttons}>**

**<TouchableOpacity**

**style={[styles.button, styles.view]}**

**onPress={() => deleteProvider(item.id)}**

**>**

**<Image**

**style={styles.icon}**

**source={{ uri: 'https://cdn-icons-png.flaticon.com/512/6861/6861362.png' }}**

**/>**

**</TouchableOpacity>**

**</View>**

**</View>**

**</View>**

**)**

**}}**

**/></View>**

**)**

**}**

**const styles = StyleSheet.create({**

**image: {**

**width: 100,**

**height: 100,**

**},**

**box: {**

**padding: 20,**

**marginTop: 5,**

**marginBottom: 5,**

**backgroundColor: 'white',**

**flexDirection: 'row',**

**},**

**boxContent: {**

**flex: 1,**

**flexDirection: 'column',**

**alignItems: 'flex-start',**

**marginLeft: 10,**

**},**

**title: {**

**fontSize: 18,**

**color: '#151515',**

**},**

**description: {**

**fontSize: 15,**

**color: '#646464',**

**},**

**view: {**

**backgroundColor: '#eee',**

**},**

**addButton: {**

**backgroundColor: '#07B0A8',**

**paddingHorizontal: 20,**

**paddingVertical: 10,**

**flexDirection: 'row',**

**alignItems: 'center',**

**justifyContent: 'center',**

**borderRadius: 5,**

**marginTop: 10, // Marge en haut**

**marginLeft: 200, // Marge à gauche**

**marginRight:10**

**},**

**addButtonText: {**

**color: 'white',**

**marginLeft: 10,**

**}, imageContainer: {**

**alignItems: 'center',marginLeft:30,**

**marginVertical: 10,width:'80%'**

**},**

**buttons: {**

**flexDirection: 'row',**

**},**

**button: {**

**height: 35,**

**flexDirection: 'row',**

**justifyContent: 'center',**

**alignItems: 'center',**

**borderRadius: 10,**

**width: 50,**

**marginRight: 5,**

**marginTop: 5,**

**},**

**icon: {**

**width: 20,**

**height: 20,**

**},**

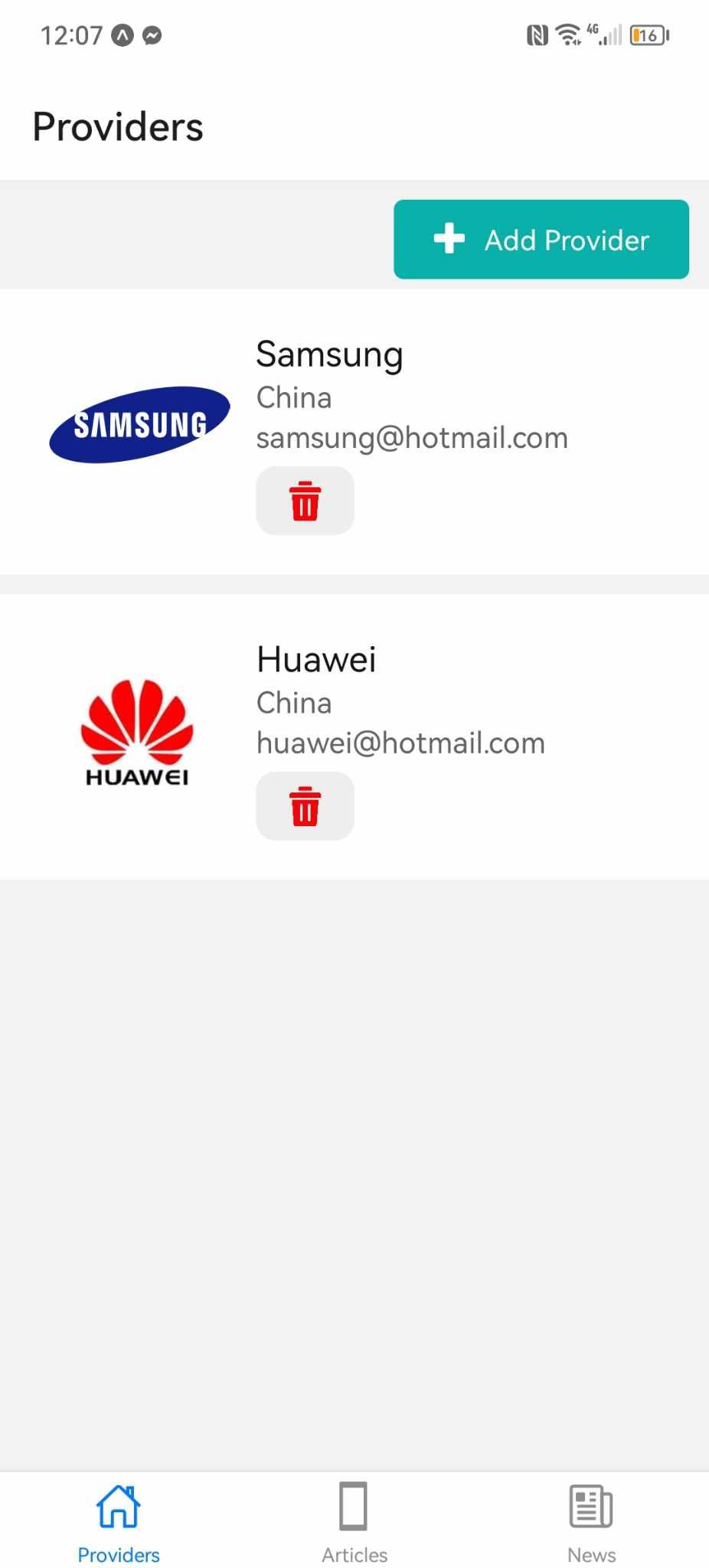
**view: {**

**backgroundColor: '#eee',**

**},**

**})**

**export default HomeScreen;**

****

On peut ajouter une fonction de **déconnexion** (logout) dans l'en-tête du TabNavigator

**import { StatusBar } from 'expo-status-bar';**

**import {**

**StyleSheet,**

**Button,TouchableOpacity**

**} from 'react-native'**

**import {createBottomTabNavigator } from '@react-navigation/bottom-tabs';**

**import HomeScreen from '../screens/HomeScreen';**

**import Ionicons from 'react-native-vector-icons/Ionicons';**

**import NewScreen from '../screens/NewScreen';**

**import ArticleScreen from '../screens/ArticleScreen';**

**import addProviderScreen from '../screens/AddProviderScreen';**

**import { createNativeStackNavigator } from '@react-navigation/native-stack';**

**import { useNavigation } from '@react-navigation/native'**

**import authtoken from '../service/authtoken';**

**const tab = createBottomTabNavigator();**

**const Stack = createNativeStackNavigator();**

**const HomeStack = () => (**

**<Stack.Navigator>**

**<Stack.Screen name="Home" component={HomeScreen} options={{**

**headerShown: false**

**}} />**

**<Stack.Screen**

**name="Add Provider"**

**component={addProviderScreen}**

**options={({route}) => ({**

**title: route.params?.title,**

**headerStyle: {**

**backgroundColor: '#DFF8F7',**

**}**

**})}**

**/>**

**</Stack.Navigator>**

**);**

**export default function TabNavigator() {**

**const navigation=useNavigation();**

**const logout = () => {**

**authtoken.logout();**

**navigation.navigate('Login');**

**};**

**const LogoutButton = () => (**

**<TouchableOpacity onPress={logout} style={{ marginRight: 10 }}>**

**<Ionicons name="log-out-outline" size={25} color="#000" />**

**</TouchableOpacity>**

**);**

**return (**

**<tab.Navigator**

**screenOptions={({ route }) => ({**

**tabBarIcon: ({ focused, color, size }) => {**

**let iconName;**

**if (route.name == "Home") { iconName = "home-outline"; }**

**else if (route.name == "Articles") { iconName = "phone-portrait-sharp"; }**

**else if (route.name == "News") { iconName = "newspaper-outline"; }**

**return (**

**<Ionicons**

**name={iconName}**

**color={color}**

**size={size}**

**/>**

**);**

**},**

**})}>**

**<tab.Screen name="Home" component={HomeStack} options={{**

**title: 'Providers',**

**headerRight: () => <LogoutButton />**

**}}/>**

**<tab.Screen name="Articles" component={ArticleScreen} options={{**

**title: 'Articles',**

**headerRight: () => <LogoutButton />**

**}}/>**

**<tab.Screen name="News" component={NewScreen} options={{**

**title: 'News',**

**headerRight: () => <LogoutButton />**

**}}/>**

**</tab.Navigator>**

**);**

**}**

**const styles = StyleSheet.create({**

**container: {**

**flex: 1,**

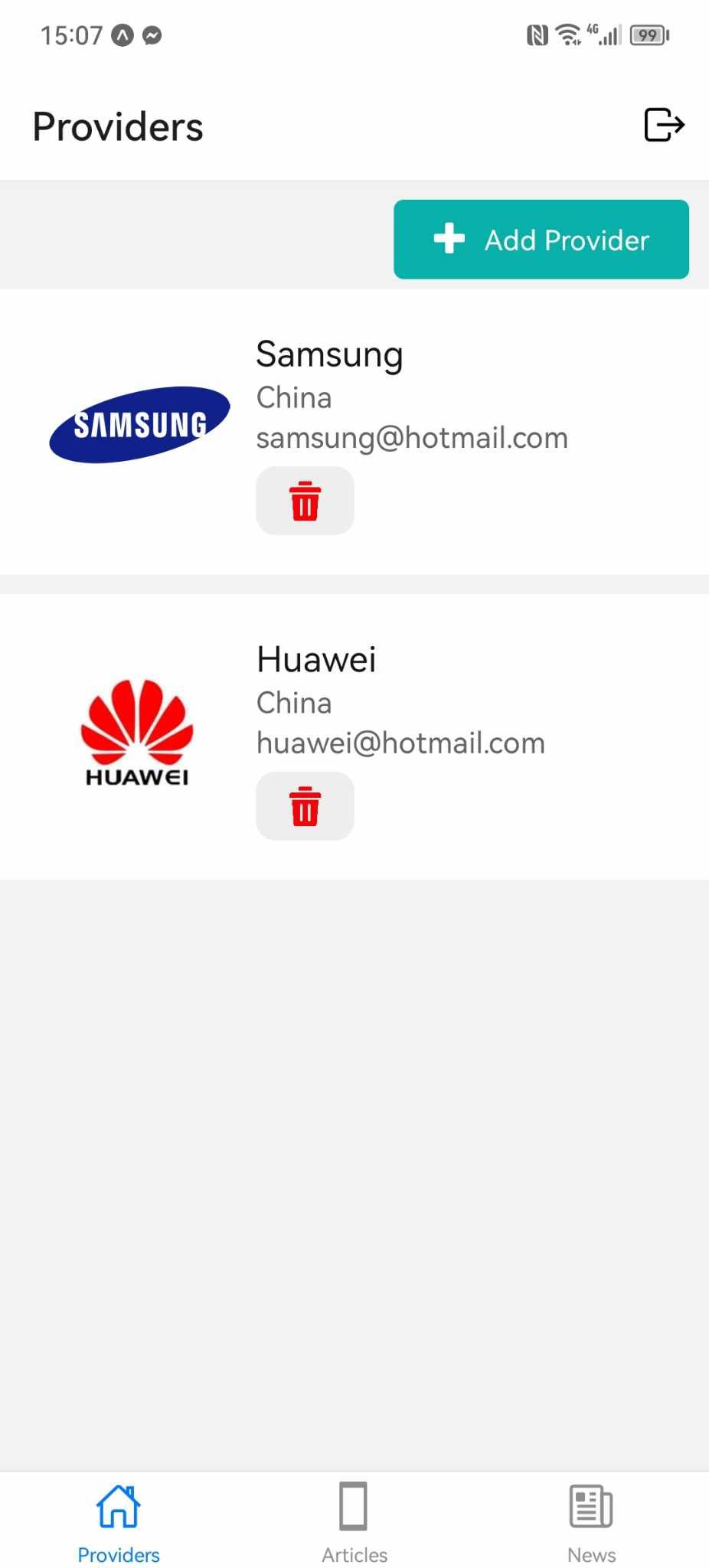
**backgroundColor: '#fff',**

**alignItems: 'center',**

**justifyContent: 'center',**

**},**

**});**

****

**Build APK**

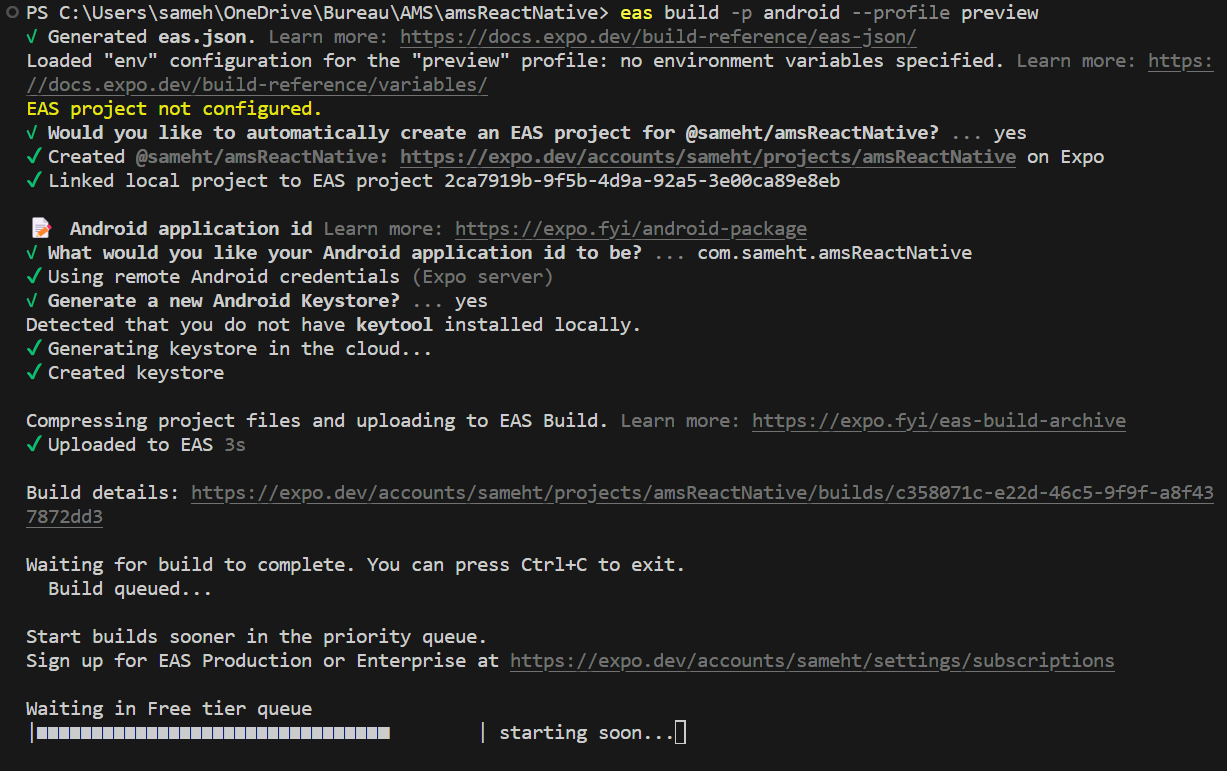
Il faut avoir un compte sur<https://expo.dev/>

npm install -g eas-cli

**eas-cli** est un outil essentiel pour les développeurs utilisant Expo pour gérer le processus de construction et de distribution de leurs applications React Native.

eas build -p android --profile preview

Cette commande lance le processus de construction d'une application pour la plateforme Android en utilisant Expo Application Services.



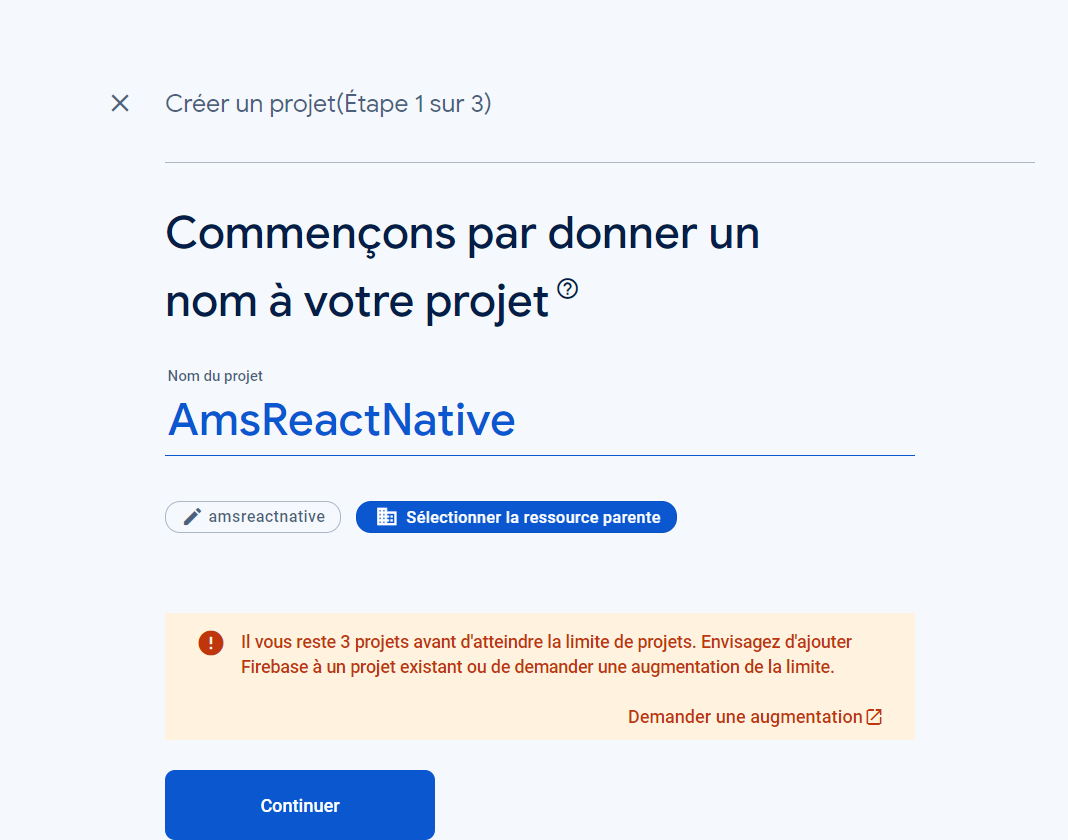
**Push notifications**

<https://firebase.google.com/>

On doit utiliser **Firebase** est une plateforme de développement d'applications mobiles et web développée par Google.Elle offre une variété de services et d'outils pour aider les développeurs à créer, améliorer et faire évoluer leurs applications plus rapidement et plus efficacement.

On doit utiliser **Firebase Cloud Messaging (FCM)** : Un service de messagerie qui permet d'envoyer des notifications et des messages à des utilisateurs sur Android, iOS et les applications web.

1-Création d’un nouveau projet

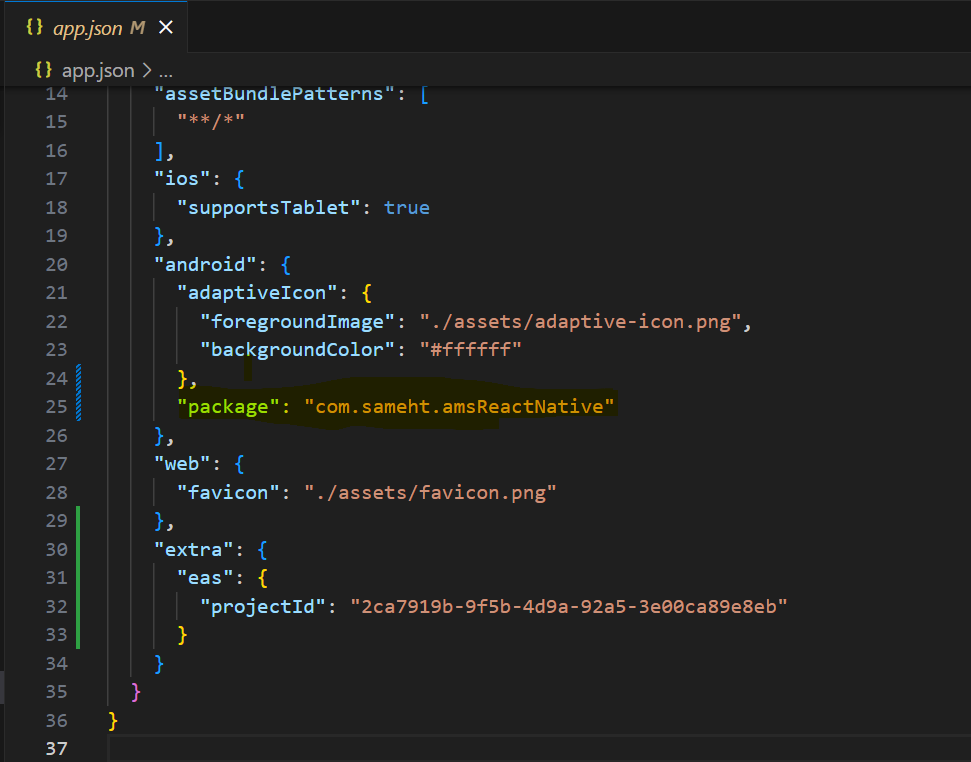


On doit ajouter une application Android





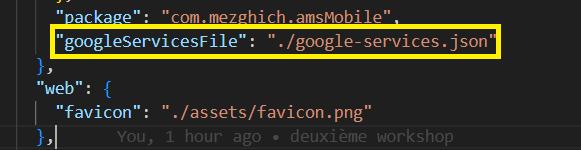
Le nom du package doit être le même dans le fichier **app.json**.

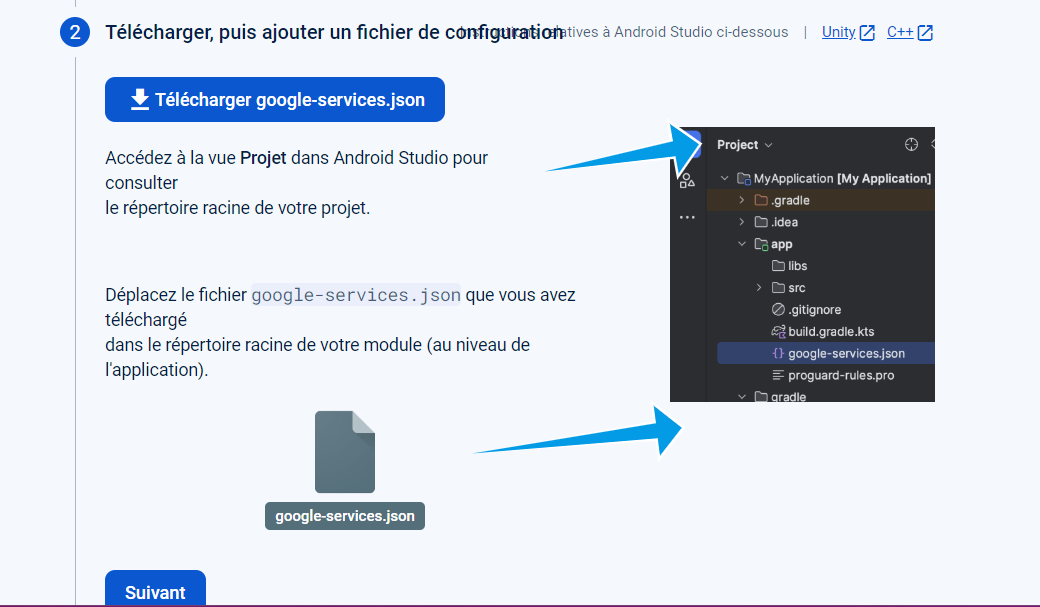


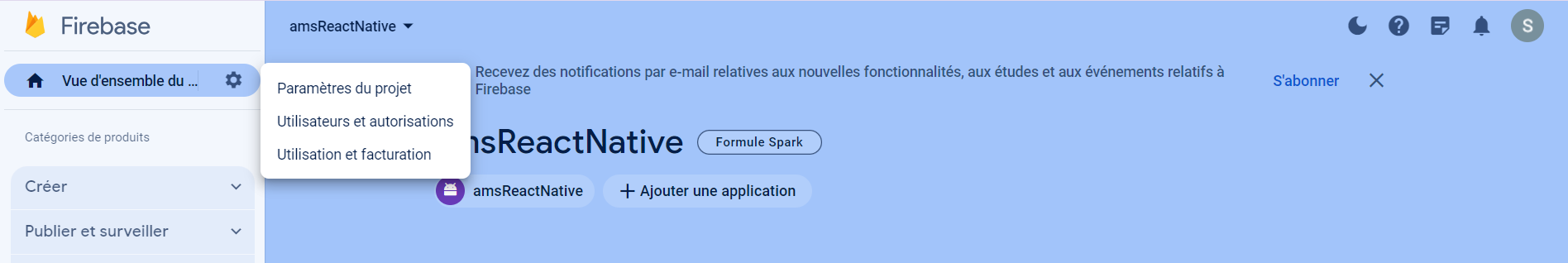
Ensuite, il faut télécharger le fichier **google-services.json** et l'ajouter à votre projet.

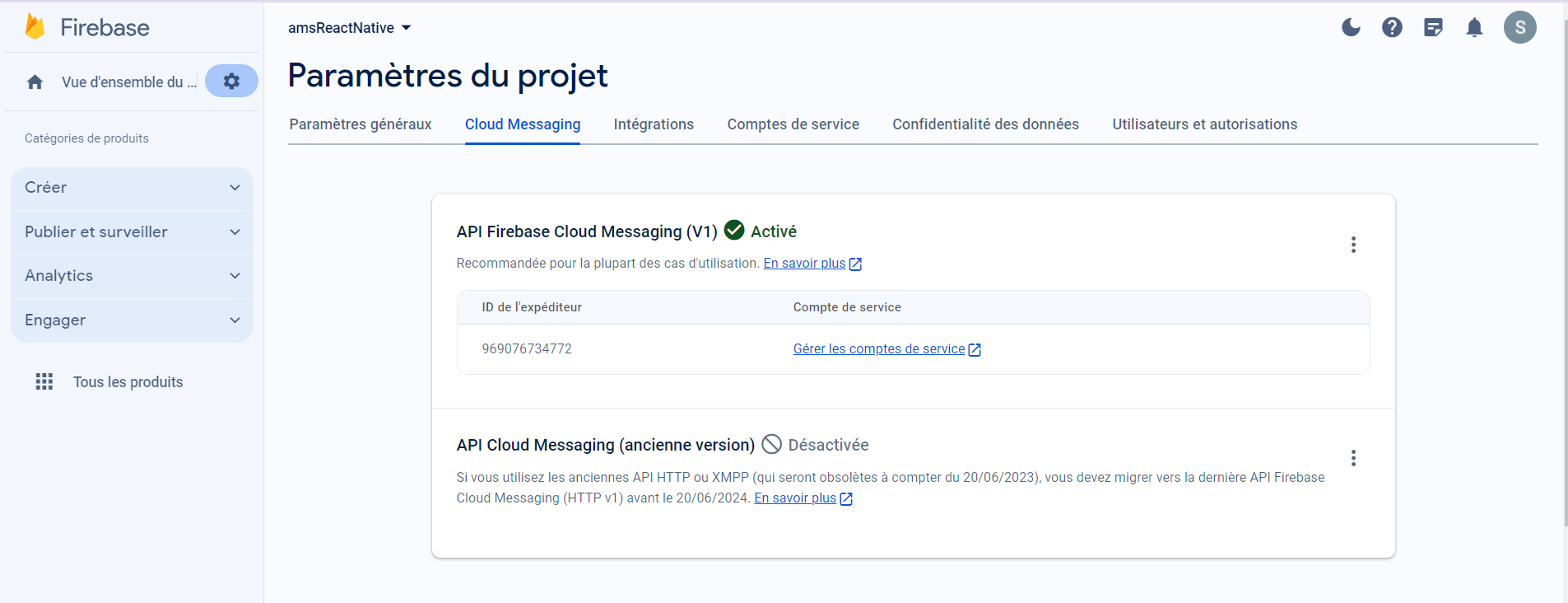
Ajouter cette ligne sous le package

"googleServicesFile": "./google-services.json"

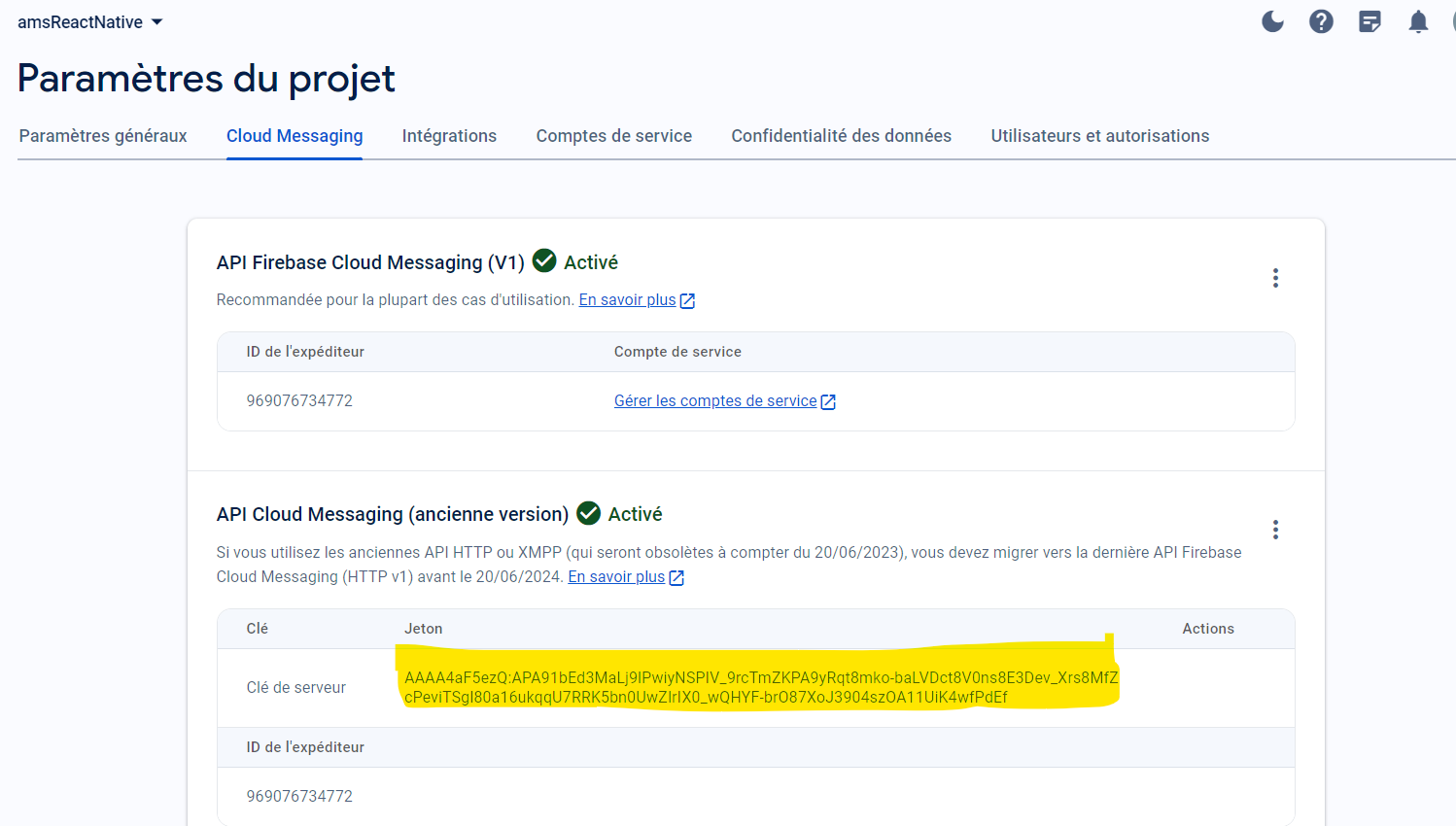
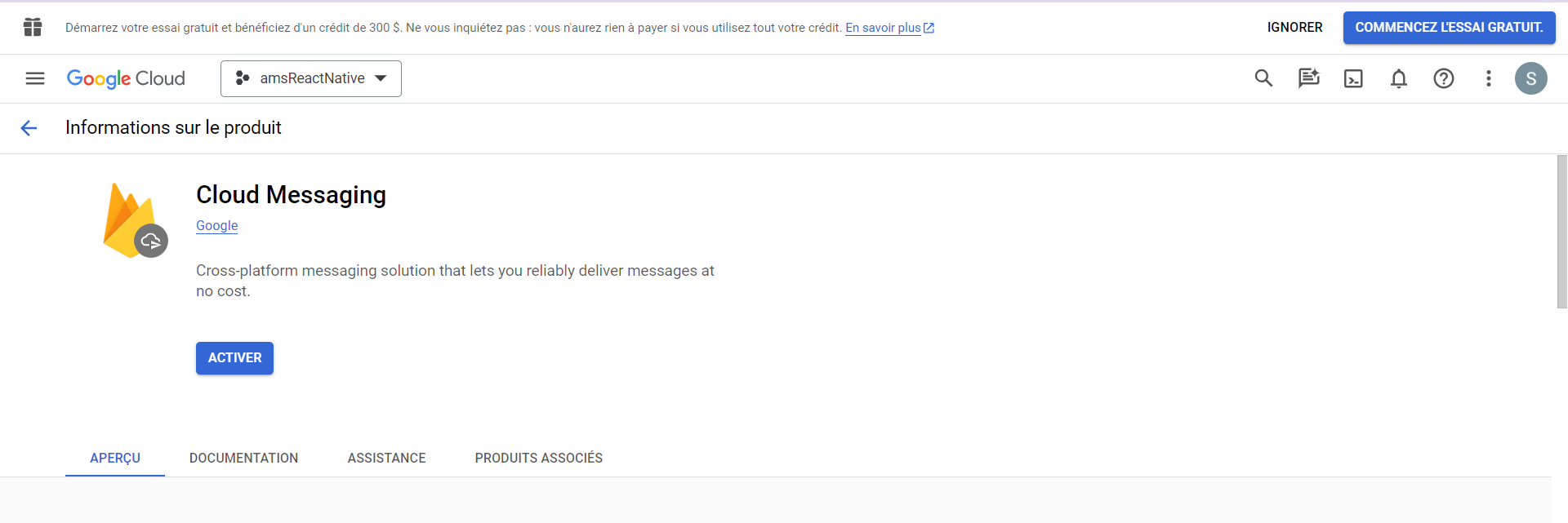
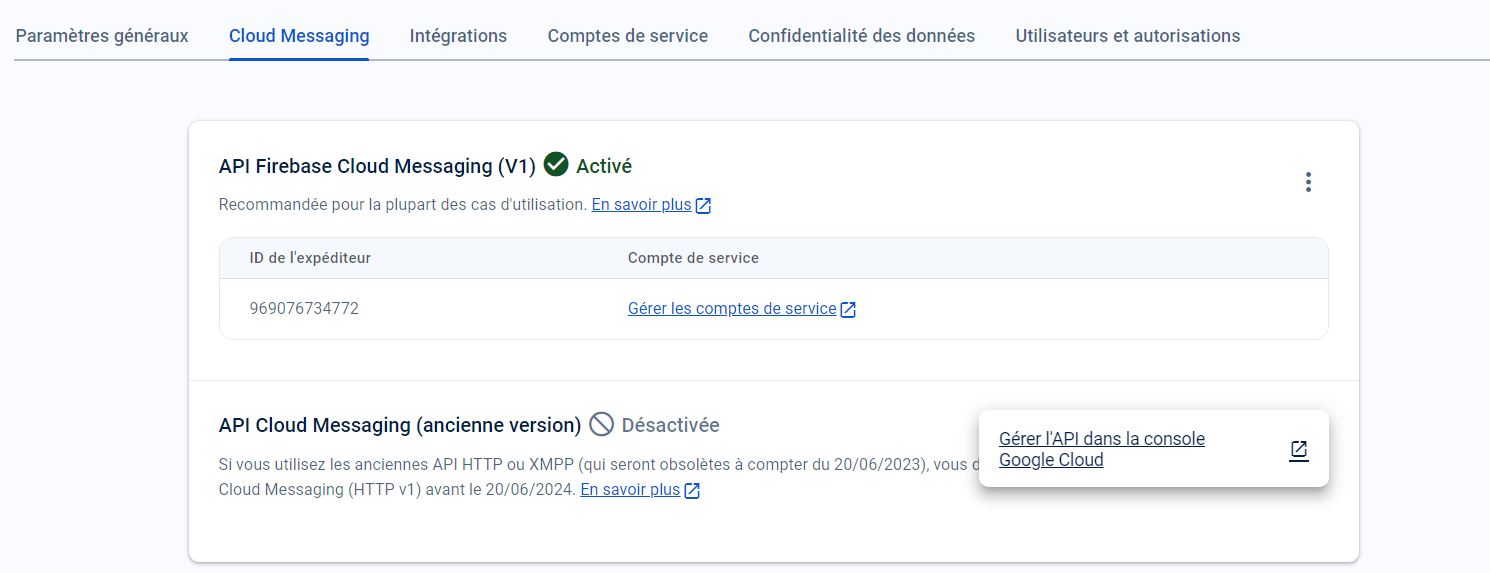






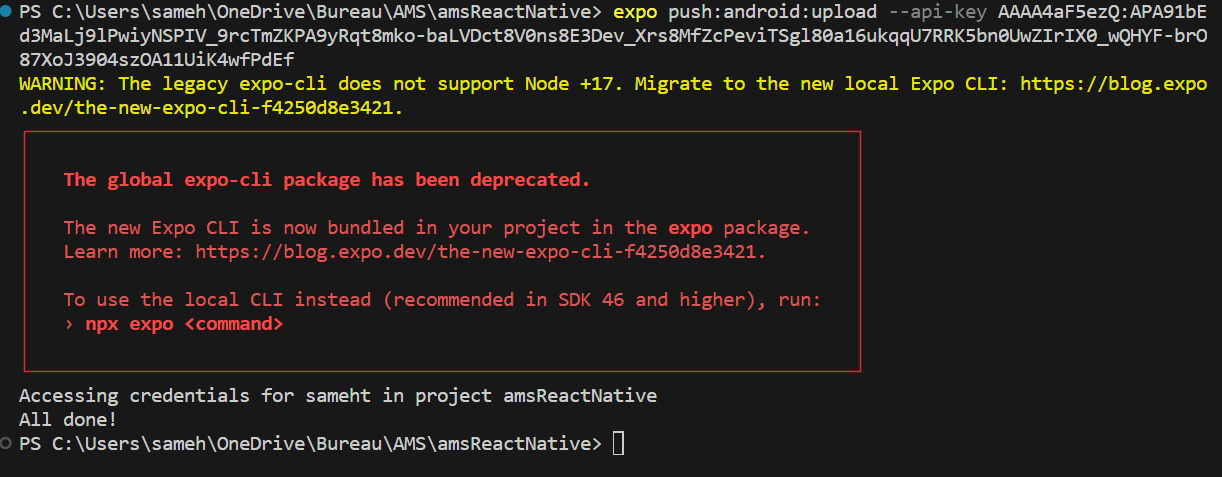


Maintenant, il est nécessaire d'activer l'API Cloud Messaging pour générer une clé de serveur.



Enfin on doit exécuter cette commande

expo push:android:upload --api-key AAAA4aF5ezQ:APA91bEd3MaLj9lPwiyNSPIV\_9rcTmZKPA9yRqt8mko-baLVDct8V0ns8E3Dev\_Xrs8MfZcPeviTSgl80a16ukqqU7RRK5bn0UwZIrIX0\_wQHYF-brO87XoJ3904szOA11UiK4wfPdEf



=>On doit exécuter deux commande

npm install expo-device

fournit des informations sur l'appareil sur lequel votre application s'exécute

npm install expo-notifications

offre une API complète pour gérer les notifications push dans les applications Expo. Il permet de programmer l'envoi de notifications, de les recevoir, et de gérer les permissions liées aux notifications.

On doit ajouter cette fonction dans **HomeScreen**

import \* as Device from 'expo-device';

import \* as Notifications from 'expo-notifications';

**Ajouter** const [devise,setDevise] = useState("")

Ensuite

**async function registerForPushNotificationsAsync() {**

**let token;**

**if (Device.isDevice) {**

**const { status: existingStatus } = await Notifications.getPermissionsAsync();**

**let finalStatus = existingStatus;**

**if (existingStatus !== 'granted') {**

**const { status } = await Notifications.requestPermissionsAsync();**

**finalStatus = status;**

**}**

**if (finalStatus !== 'granted') {**

**alert('Failed to get push token for push notification!');**

**return;**

**}**

**token = (await Notifications.getExpoPushTokenAsync()).data;**

**setDevise(token)**

**const id = await asyncStorage.getItem("id");**

**axios.post(**

**"https://ams.smart-it-partner.com/api/push\_notives",**

**{ token: token, user: id },**

**{ headers: { 'Content-Type': 'application/ld+json' } }**

**);**

**} else {**

**alert('Must use physical device for Push Notifications');**

**}**

**if (Platform.OS === 'android') {**

**Notifications.setNotificationChannelAsync('default', {**

**name: 'default',**

**importance: Notifications.AndroidImportance.MAX,**

**vibrationPattern: [0, 250, 250, 250],**

**lightColor: '#FF231F7C',**

**});**

**}**

**return token;**

**}**

**useEffect(() => {**

**registerForPushNotificationsAsync();**

**Notifications.setNotificationHandler({**

**handleNotification: async () => ({**

**shouldShowAlert: true,**

**shouldSetBadge: true,**

**}),**

**});**

**}, []);**

***Bon travail !***