**Cinquime Partie : Push notification**

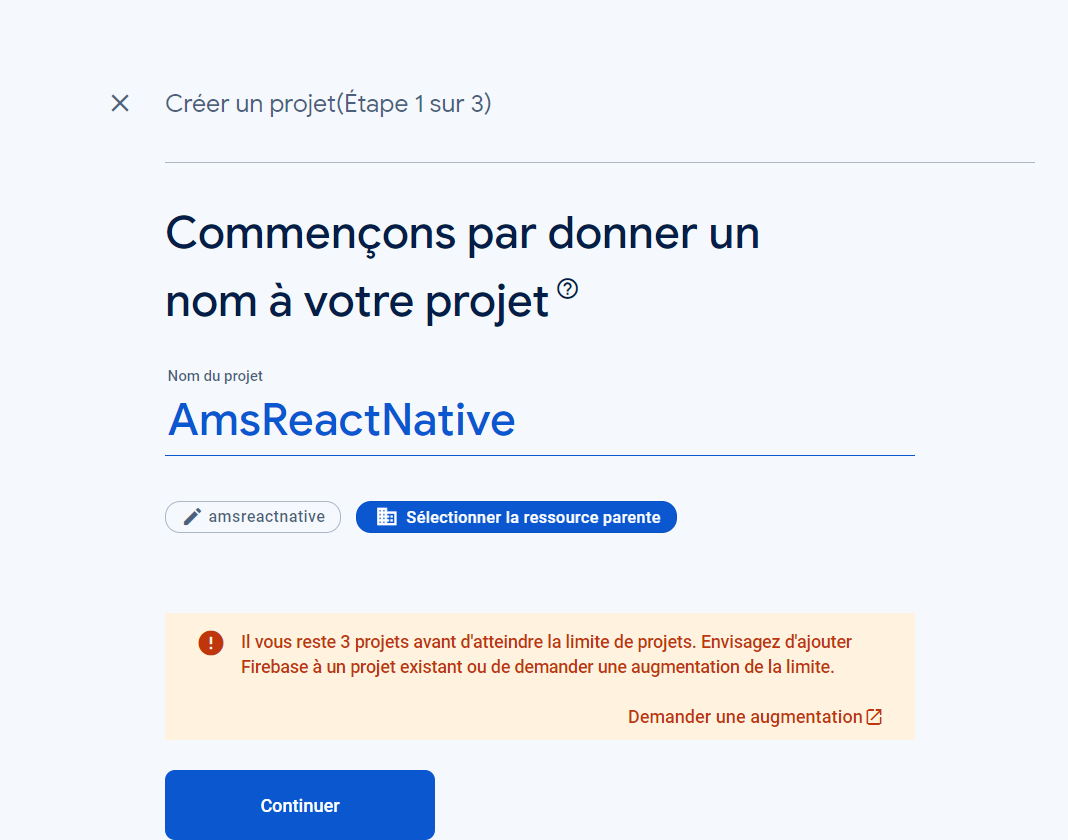
**Push notifications**

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

**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

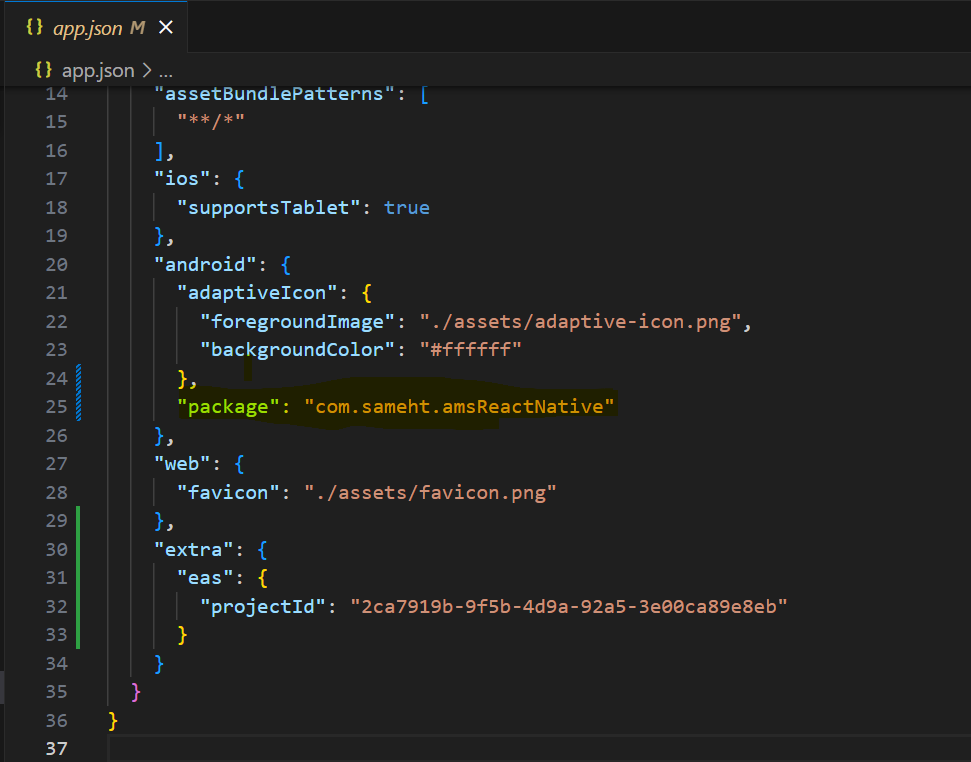


On doit ajouter une application Android

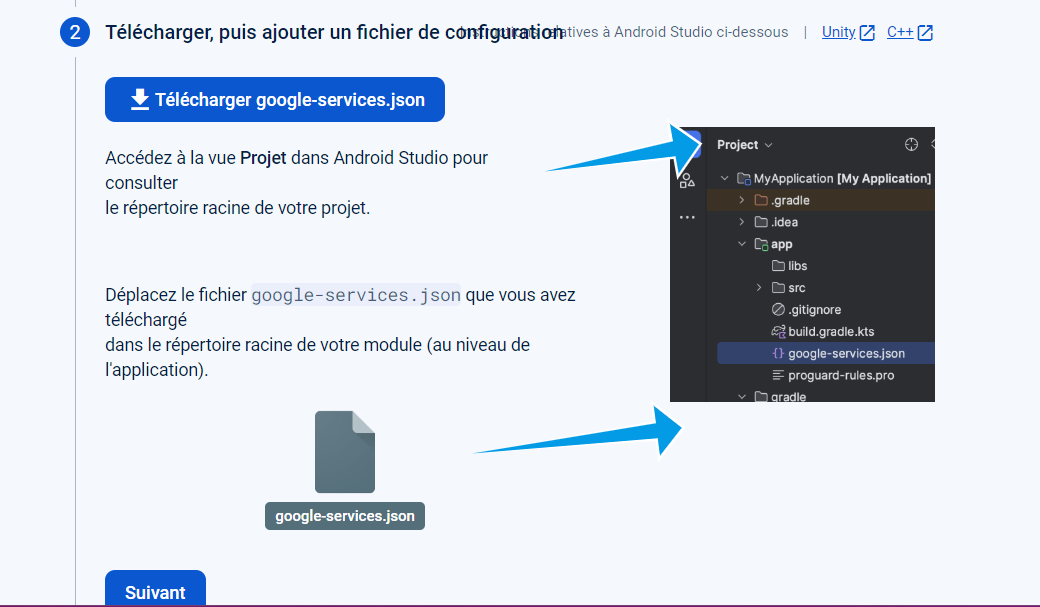


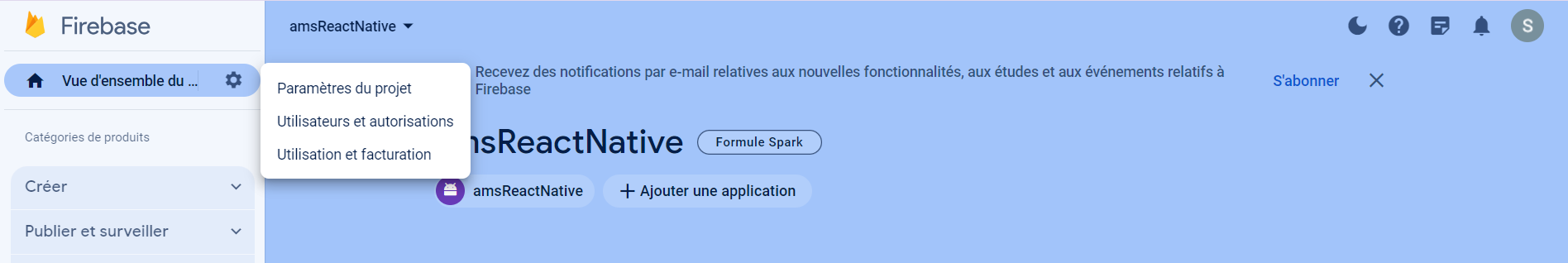


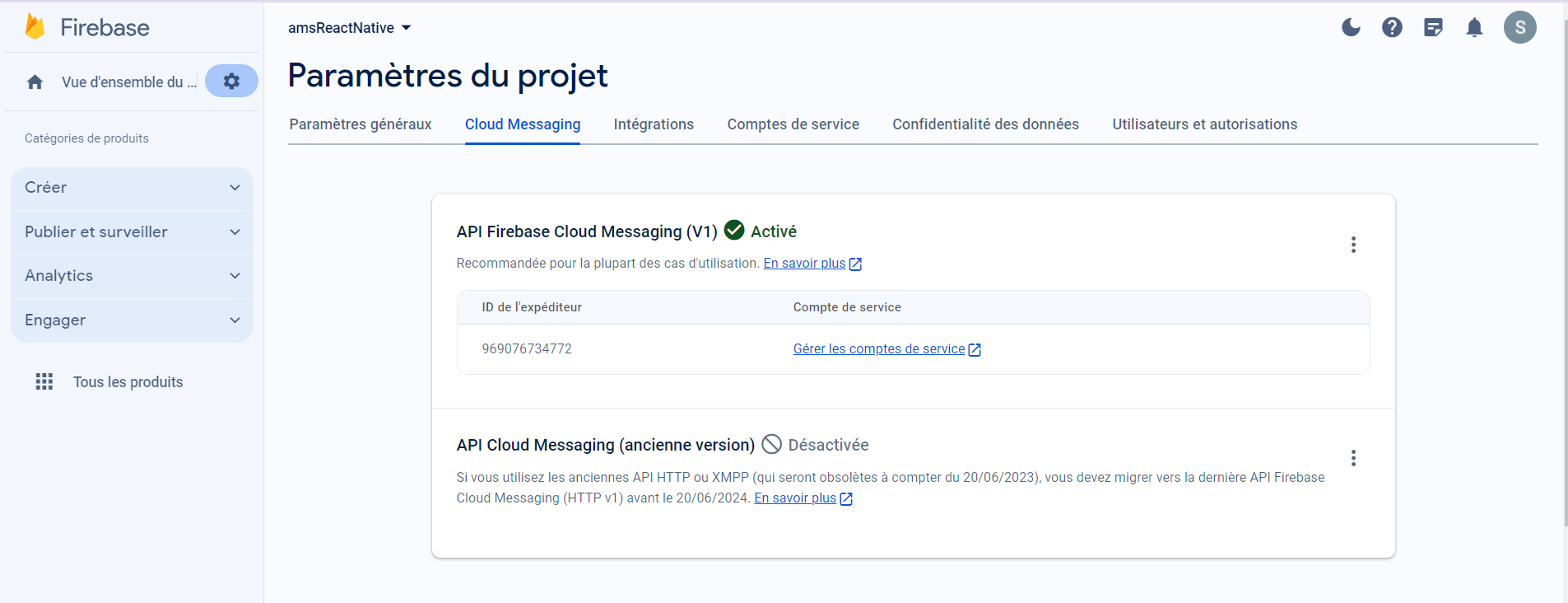
Le nom du package doit être le même dans le fichier **app.json**.



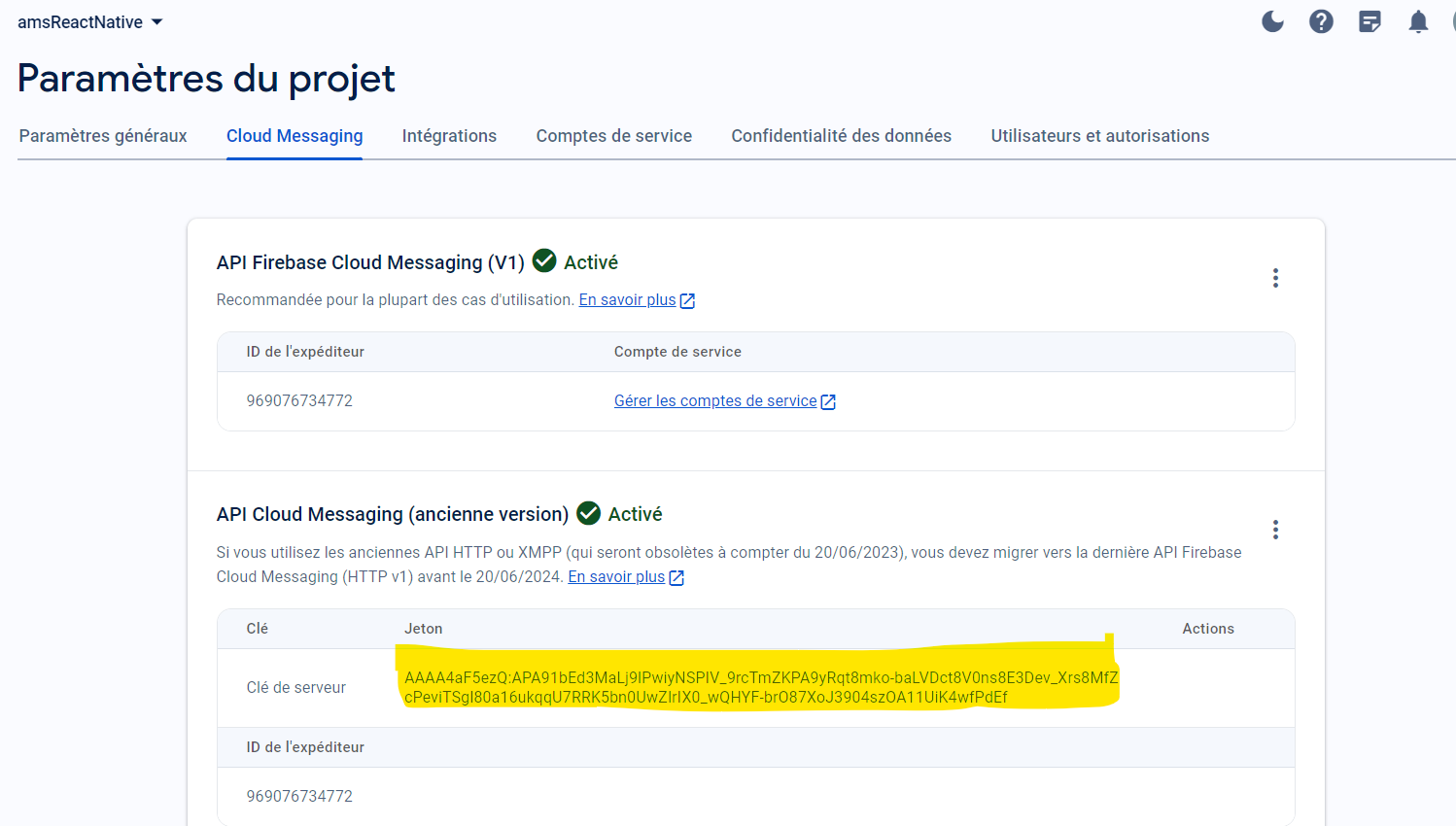
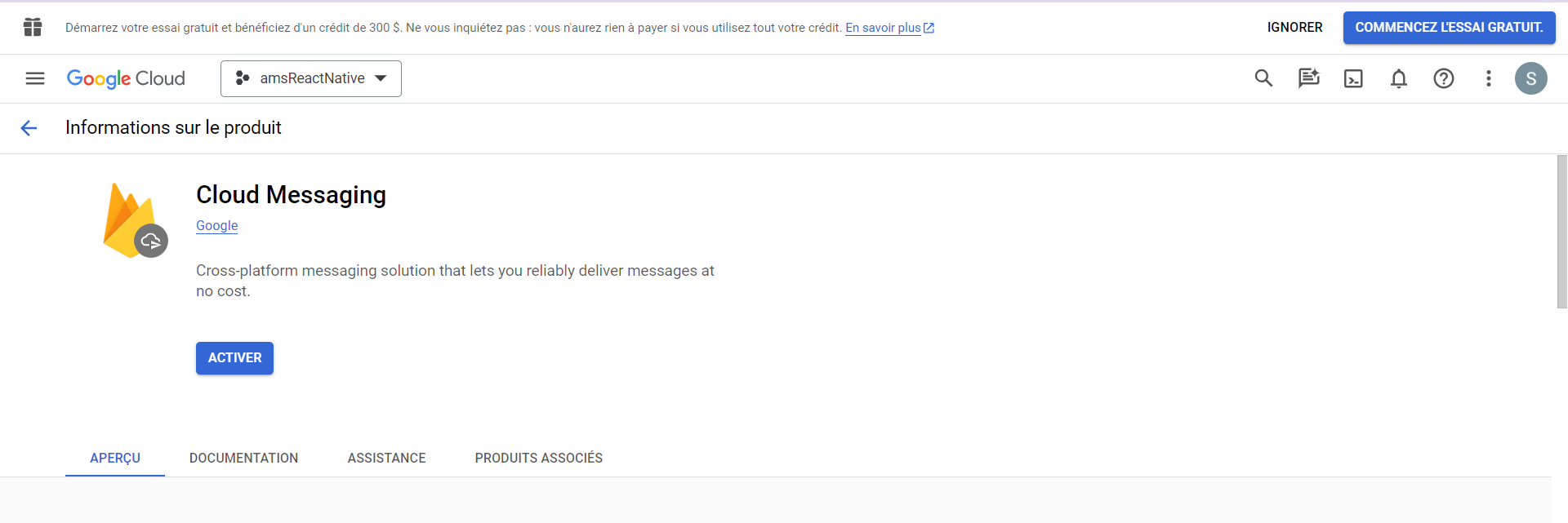
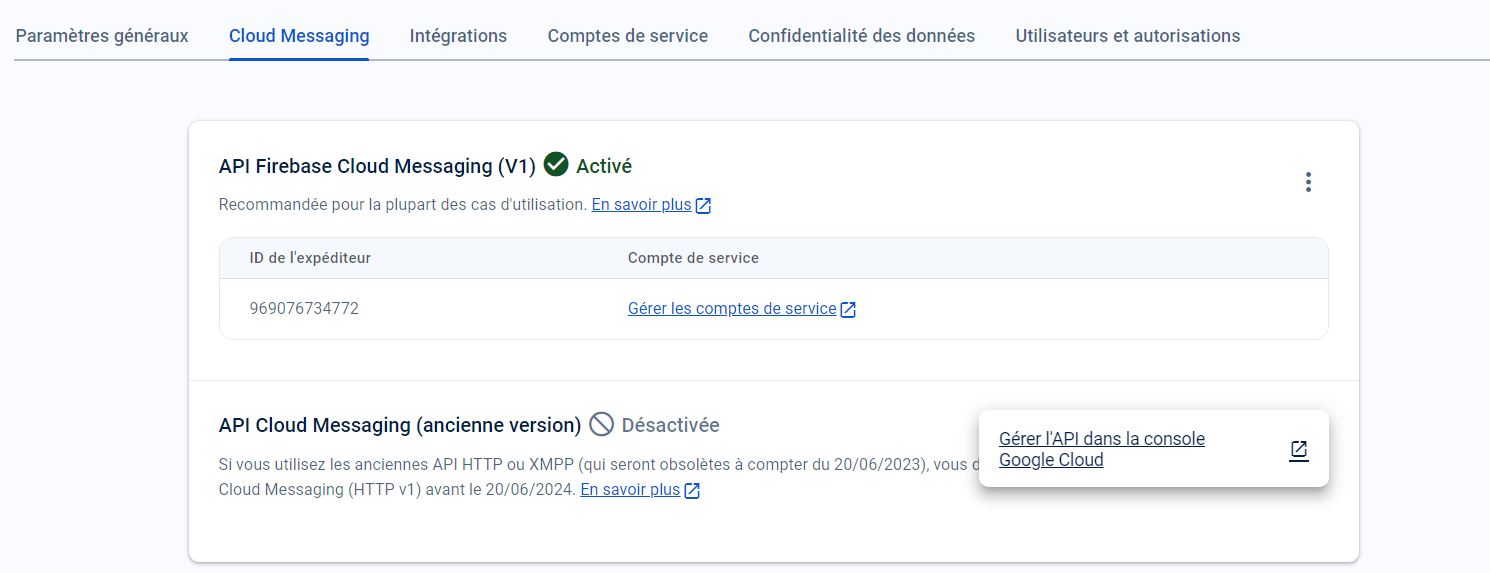
Ensuite, il faut télécharger le fichier **google-services.json** et l'ajouter à votre projet.





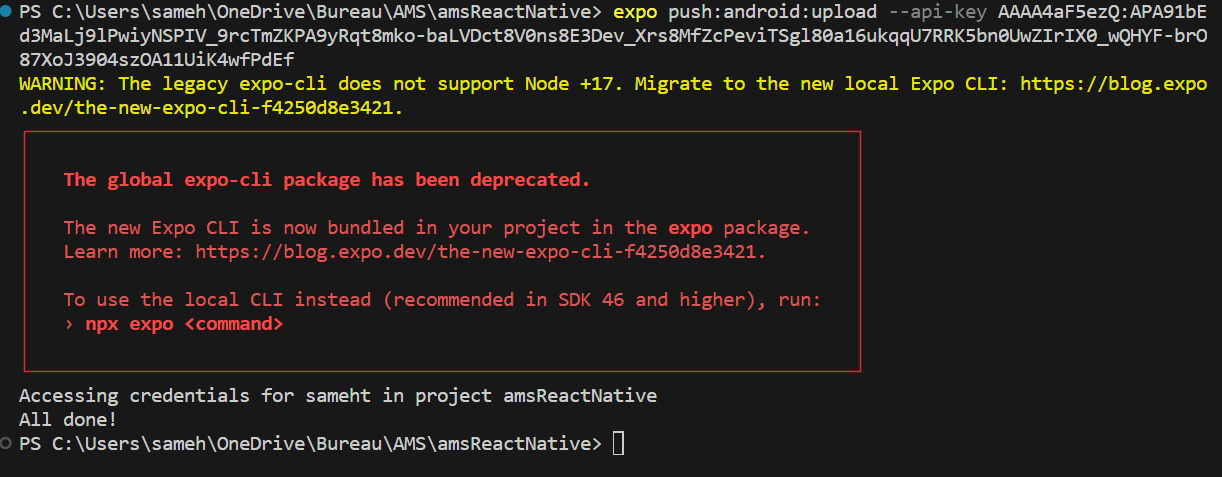


Maintenant, il est nécessaire d'activer l'API Cloud Messaging pour générer une clé de serveur.



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

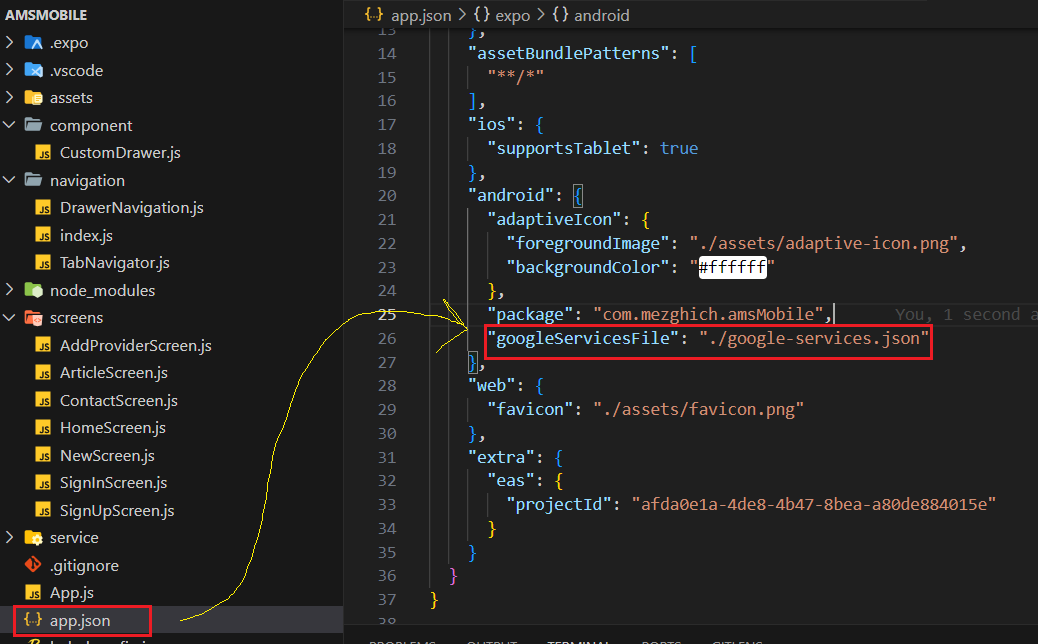
fournit des informations sur l'appareil sur lequel votre application s'exécute

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.

Ajouter cette instruction dans votre app.json file

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



**Push notifications vers lien spécifique**

On doit modifier **registerForPushNotificationsAsync** dans **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';

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

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

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 editProvider = (idprovider) => {

        navigation.navigate("Add Provider", { idprovider });

    };

    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: 'List Providers', params: { refresh: true } }],

        });

    }

    // push notification

    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;

            //setRef(token)

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

            const id = "15";

            console.log("token :" + token);

            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');

            // cas emulateur android

           // setRef(token)

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

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

            const id = "15";

            console.log("token :" + token);

            axios.post(

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

                { token: token, user: id },

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

            );

        }

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

            Notifications.setNotificationChannelAsync('default', {

                name: 'default',

                importance: Notifications.AndroidImportance.MAX,

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

                lightColor: '#FF231F7C',

            });

        }

        return token;

    }

    useEffect(() => {

        registerForPushNotificationsAsync(); //appel pour s'enregister

        // Configure le comportement de la notification lorsque l'application est ouverte

        Notifications.setNotificationHandler({

            handleNotification: async () => ({

                shouldShowAlert: true, // Affiche une alerte même si l'app est ouverte

                shouldPlaySound: true,

                shouldSetBadge: false,

            }),

        });

        // Gestionnaire pour les interactions avec les notifications

        const responseSubscription = Notifications.addNotificationResponseReceivedListener(response => {

            const data = response.notification.request.content.data;

            // Vérifie si l'application était en premier plan au moment où l'utilisateur a interagi avec la notification

            if (response.notification.request.trigger.type === 'push' && data.screen && navigation) {

                // Naviguez vers l'écran spécifié

                navigation.navigate(data.screen, {

                    title: data.title,

                    id: data.id

                })

            }

        });

        return () => {

            responseSubscription.remove();

        };

    }, [navigation]);

    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={() => editProvider(item.id)}

                                    >

                                        <Image

                                            style={styles.icon}

                                            source={{ uri: 'https://cdn-icons-png.flaticon.com/512/2429/2429794.png' }}

                                        />

                                    </TouchableOpacity>

                                    <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;

****

***Bon travail !***