# Travail final

## Objectifs :

* + Lire du JSON provenant d'un serveur externe ( Annexe 6 )
  + Utiliser React Navigation pour naviguer entre différents écrans ( Annexe 7 )
  + Présenter des informations en utilisant des composants ReactNative ( FlatList, Image, Button, etc )

Le travail, relativement simple étant donné le temps qu'il nous reste, sera de présenter des informations et des images présentes dans le web API de Spotify à partir d'un artiste que vous appréciez en utilisant la technologie React Native

Un grand nombre de données ouvertes sont présentes dans cet API ; cliquez sur les liens suivants :

<https://developer.spotify.com/documentation/web-api/reference/get-an-artist>

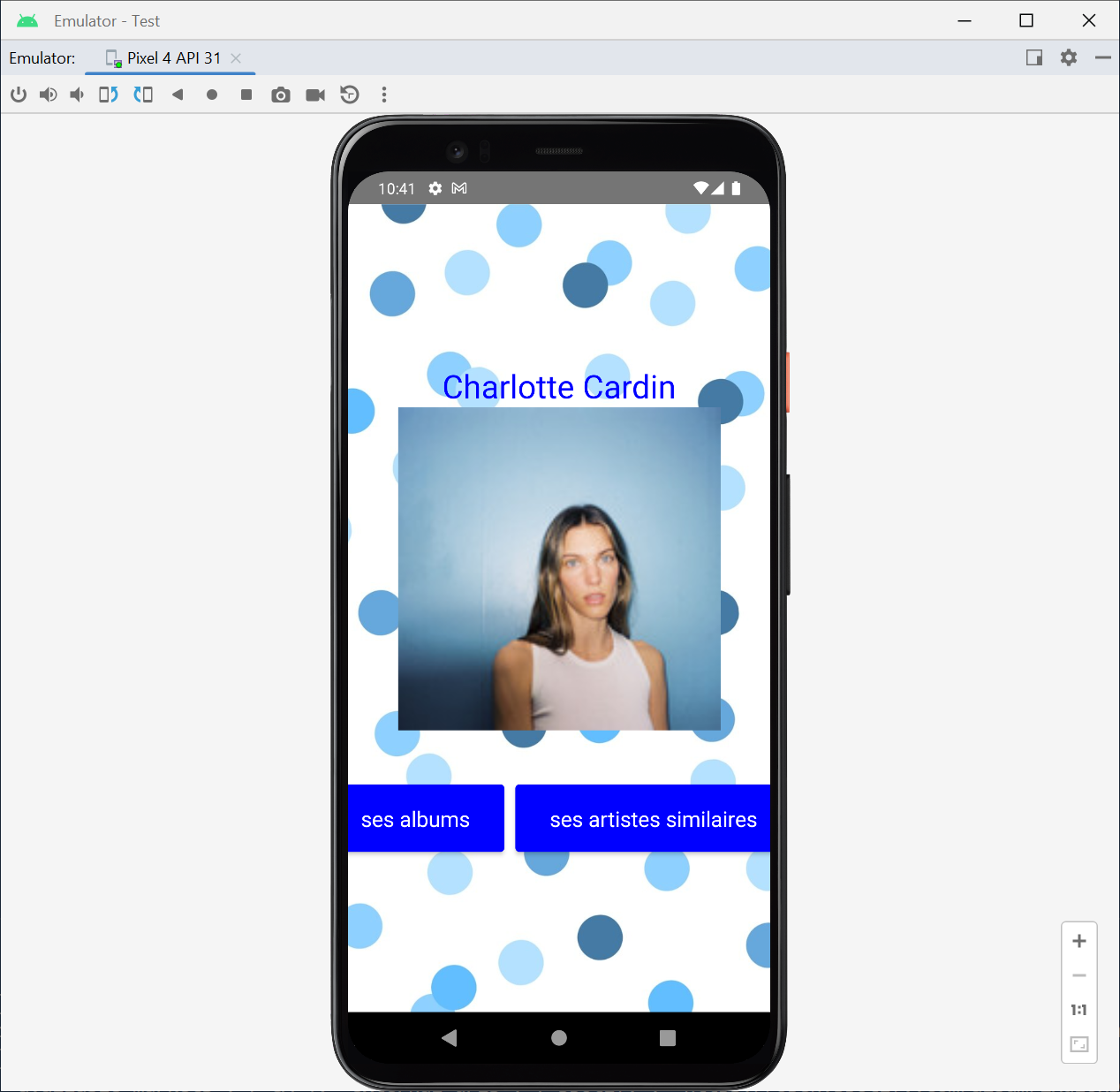
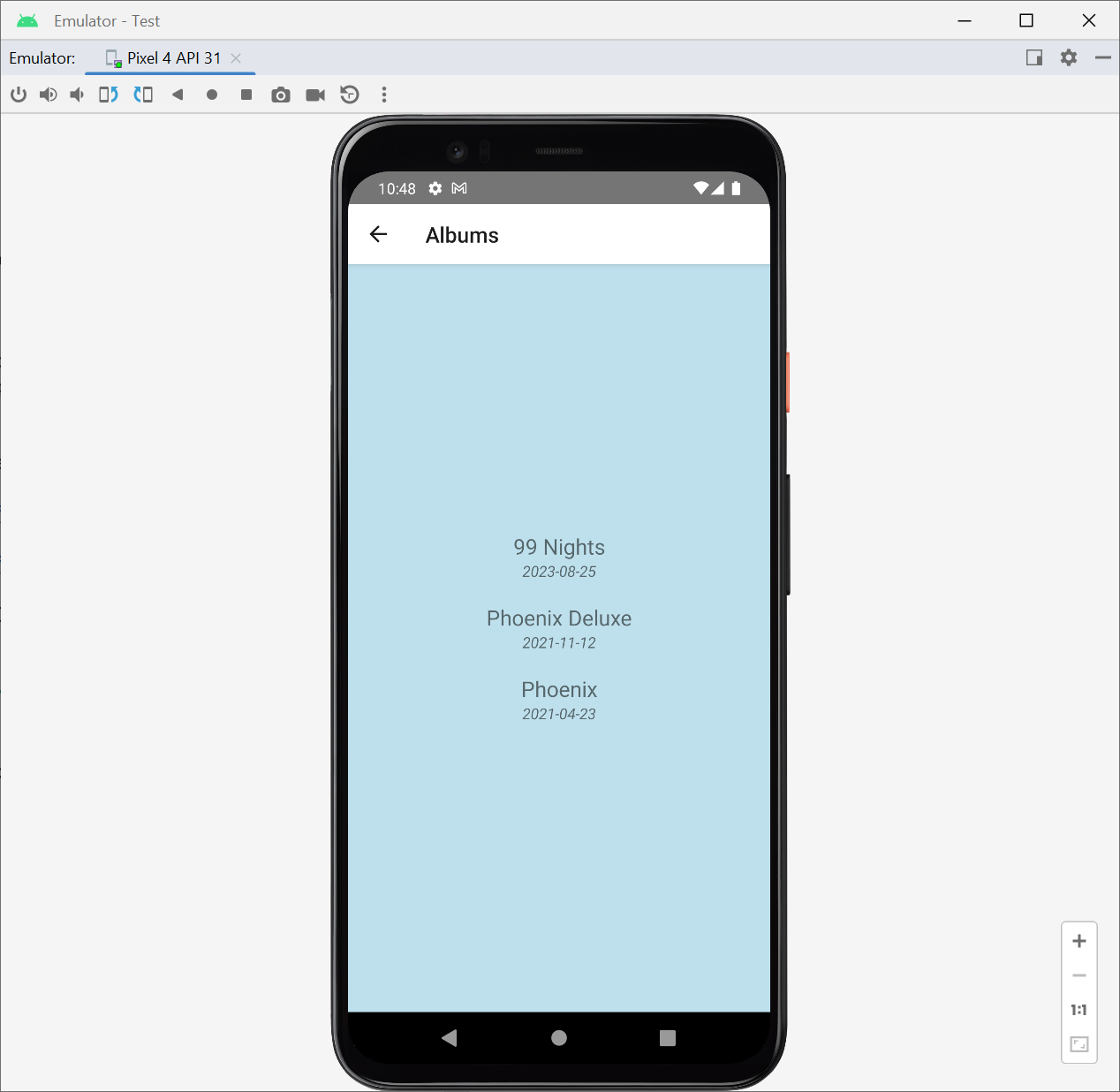
<https://developer.spotify.com/documentation/web-api/reference/get-an-artists-albums>

Vous pouvez tester avec un artiste en utilisant Spotify. ENDPOINT sera l'adresse à passer à notre fetch comme on a fait à l'annexe 6.

## Vous pouvez faire plusieurs requêtes de manière à afficher :

* Une ou plusieurs images de cet artiste
* Des boutons nous amenant, à partir d'un écran d'accueil, vers d'autres écrans présentant des informations sur les albums, chansons ou autres statistiques sous forme de FlatList et / ou de composants fonctionnels que vous aurez conçus

## Exemple ( choisissez d'autres requêtes et une autre artiste )

*Page d'accueil et lorsqu'on clique sur "Ses albums" , on les affiche avec un composant FlatList. J'ai filtré le tableau pour exclure les singles qui étaient considérés comme des albums aussi.*

* Les écrans devront être dans des fichiers distincts
* Un / des objets représentant un style doivent être utilisés de manière à rendre l'affichage agréable

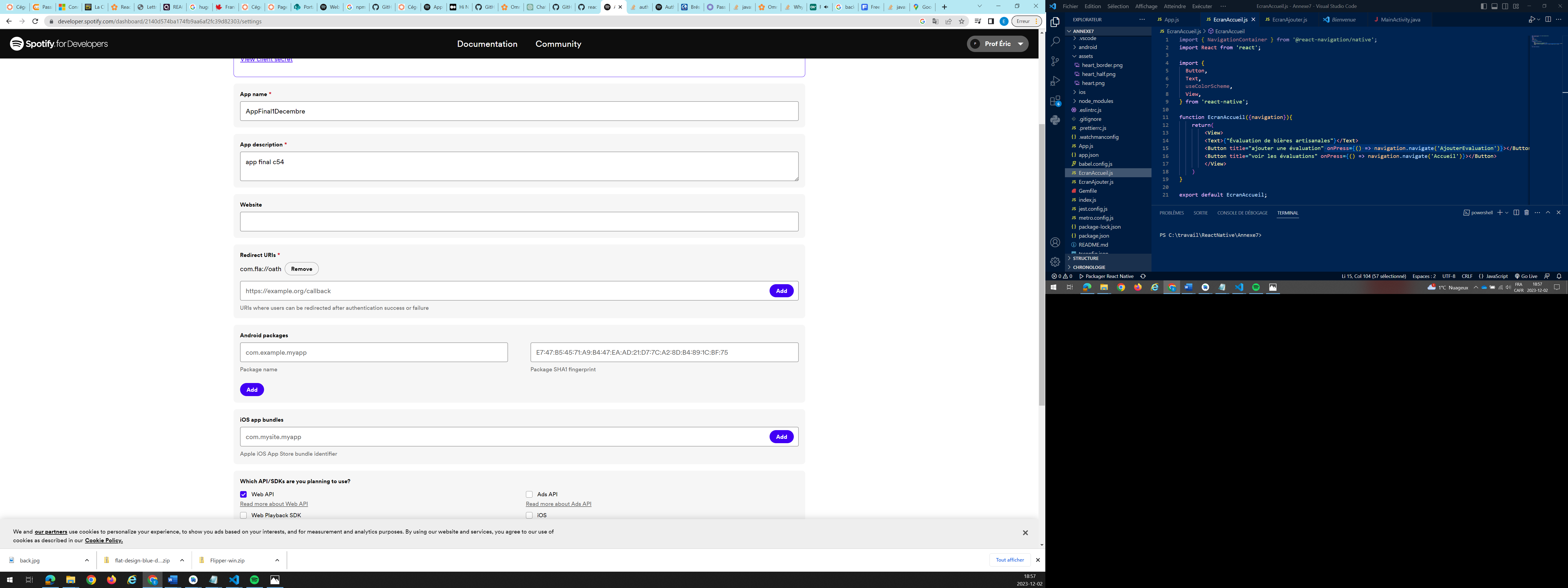
## Notes importantes

1. Autorisation

Dans ce travail, on utilisera le service OAuth 2.0 pour s'authentifier à Spotify et pour ensuite consulter les ressources JSON.

Créer une nouvelle application React Native. Notez le nom du package présent dans le dossier app/src/main/java/MainActivity pour y noter **le nom du package présent à la première ligne**

1. Créer un nouveau projet dans le Dashboard de Spotify for Developers comme dans le tp1:



* Entrez appName, app Description et RedirectUrl composé de votre **nom de package ://oath**
* Pas besoin de SHA-1
* N'oubliez pas de sauvegarder

B- On doit ensuite utiliser un bridge produit par FormidableLabs qui permettra de générer un "accessToken" qu'on ajoutera ensuite au header de nos requêtes GET pour rejoindre nos fichiers JSON.

Le site web est ici : <https://github.com/FormidableLabs/react-native-app-auth>

Je vous résume la procédure présente à <https://github.com/FormidableLabs/react-native-app-auth#getting-started>

* npm install react-native-app-auth –save pour installer
* Ajouter la propriété suivante à la partie defaultConfig du fichier  android/app/build.gradle:

android {

defaultConfig {

manifestPlaceholders = [

appAuthRedirectScheme: 'votre nom de package de votre dashboard/projet'

]

}

}

Sauvegardez…

C- Utilisez le fichier SpotifyAuth.js fourni et remplissez les clientId, clientSecret et redirectURL par vos données de votre projet dans votre Dashboard

Vous pouvez ensuite importer dans vos écrans / autres fichiers :

import { authenticateSpotify } from './SpotifyAuth';

et pouvoir appeler authenticateSpotify() afin d'obtenir une variable accessToken, qui est le fameux token dont on a besoin. N’oubliez pas les mots-clés await / async pour s'assurer de pouvoir obtenir le token en temps.

Je me suis inspirer de cet exemple : ( pas besoin de faire la partie sur le client secret )

<https://github.com/FormidableLabs/react-native-app-auth/blob/main/docs/config-examples/spotify.md>

1. On aura besoin de ce token pour l'ajouter à nos requêtes GET passé par l'API fetch en lui passant un header.

On peut passer un 2e paramètre à la fonction fetch comme ceci :

const response = await fetch('https://api.spotify.com/v1/artists/1G0YV9WooUBjrwDq0Q7EFK', { method: 'GET',

headers: {

Authorization: `Bearer ${token}`,

'Content-Type': 'application/json',

},} );

Ensuite, votre travail devrait ressembler à l'annexe 6. / 6B

1. Si vous avez besoin de passer des informations d'un écran à un autre ( pour ma part j'ai passé le token de manière à ne pas demander de me ré-identifier sur spotify à chaque écran ) , vous pouvez ajouter un deuxième paramètre à la méthode navigate que nous avons utiliser à l'annexe 7

Exemple : <https://reactnavigation.org/docs/params>

1. Si jamais vous voulez filtrer les données récupérées dans un JSON, vous pouvez utiliser la fonction filter de JavaScript : <https://masteringjs.io/tutorials/fundamentals/filter-array-of-objects>