
Rapport de projet développement mobile d'exploitation d'une API Web

11/05/2021

Table des matières

Introduction	2
I/ Présentation de l'API choisie	2
I.A/ Objectif	2
I.B/ Fonctionnalités de l'API	2
II/ Choix techniques	3
II.A/ Fonctionnalités disponibles	3
II.B/ Architecture de l'application	3
III/ Organisation	5
IV/ Conclusion	6

Introduction

Dans le cadre du module CO8SFDM0 de développement mobile de deuxième année à l'ENSC, nous avons été amenés à réaliser une application mobile devant présenter des données issues d'une API externe. Cette application devait être développée à l'aide d'Expo et basée sur le langage TypeScript afin de mettre en pratique les connaissances acquises au cours du semestre. Cette application est l'objet du présent rapport. Nous présenterons en premier lieu l'API choisie pour la réalisation de ce projet puis nous expliciterons les différents choix fonctionnels et notre organisation avant de conclure.

Pour faire fonctionner l'application, il est nécessaire de télécharger le code disponible sur le dépôt [Github](#). Une fois le dossier téléchargé il est nécessaire de lancer (dans ce fichier) les commandes "npm add expo" puis "expo start".

I/ Présentation de l'API choisie

I.A/ Objectif

Pour réaliser ce projet nous avons choisi d'exploiter l'API proposée par le site [Genius](#). Ce site recense des paroles de chansons ainsi que des explications sur le sens et le choix des paroles. Nous souhaitions donc créer à l'aide de leur API une application similaire permettant de chercher des chansons pour ensuite en afficher les paroles. Nous avons cependant rencontré un problème: l'API de Genius ne fournit pas de paroles pour des raisons légales de droits d'auteur. Cette situation est identique pour toutes les API gratuites que nous avons trouvées. Nous avons néanmoins choisi de conserver l'API de Genius et nous avons mis en place un moyen de contourner partiellement ce problème en affichant la page du site Genius correspondante à la chanson dans notre application.

I.B/ Fonctionnalités de l'API

La documentation de l'API est disponible [ici](#). Elle offre 5 grandes fonctionnalités:

- La possibilité de réaliser une recherche par titre de chanson ou par artiste. Cette recherche renvoie uniquement les 10 chansons les plus pertinentes par rapport à la recherche.
- La recherche d'un titre spécifique grâce à son identifiant. Cela permet d'obtenir plus de détails sur la chanson avec notamment une description des

circonstances de création de celle-ci et l'url de la page correspondante sur le site Genius.

- La recherche d'un artiste spécifique grâce à son identifiant afin d'obtenir les différentes informations disponibles sur celui-ci.
- La recherche d'un album spécifique grâce à son identifiant afin d'obtenir différentes informations sur celui-ci. Il n'est cependant pas possible d'obtenir les titres qui le composent.
- Il est également possible de récupérer les annotations disponibles sur les différentes pages Genius. Nous avons cependant choisi de ne pas utiliser cette fonctionnalité car il nous paraissait peu intéressant d'obtenir les annotations sur les paroles d'une chanson sans pouvoir obtenir lesdites paroles.

L'utilisation de l'API se fait grâce à un token d'identification qui nous a été donné après la création d'un compte.

II/ Choix techniques

II.A/ Fonctionnalités disponibles

Notre application possède différentes fonctionnalités s'axant sur la consultation de productions musicales :

- La recherche d'une chanson par son titre ou son artiste.
- Pour chaque chanson : l'affichage d'une page qui comprend l'artiste, l'album, une description succincte de la chanson ainsi que les paroles à travers une webview renvoyant à l'affichage de la page associée à la chanson sur le site Genius.
- Pour chaque artiste : affichage d'une brève description.
- Pour chaque album : affichage de sa date de sortie ainsi que les personnalités ayant contribué à sa création (leurs noms ainsi que leurs rôles).

II.B/ Architecture de l'application

La navigation est gérée par React Navigation et l'application est organisée en 4 écrans distincts:

- L'écran de recherche de chansons, il s'agit de l'écran d'accueil de l'application. Cet écran est constitué d'une barre de recherche permettant la saisie d'une chanson ou d'un artiste. Une fois cette recherche validée les 10 résultats les plus pertinents s'afficheront sous forme de liste sur ce même écran. Chaque résultat permet l'accès au détail de la chanson ainsi qu'à l'artiste.

- L'écran de détails d'une chanson qui est accessible après une recherche. Cet écran affiche l'artiste ainsi que l'album associé à la chanson. Si une description supplémentaire est disponible sur la chanson, celle-ci est également affichée. Enfin un composant [WebView](#) permet d'afficher la page Genius associée à la chanson et donc de pouvoir lire les paroles de celles-ci. Cependant la **WebView est disponible uniquement sur Android ou IOS**, le composant ne fonctionne pas sur navigateur web car le site n'autorise pas son affichage dans une autre page web.
- L'écran de détails d'un album qui est accessible depuis l'écran de détails d'une des chansons de l'album. Cet écran affiche les différentes informations disponibles sur celui-ci et notamment toutes les personnes ayant participé à la réalisation de celui-ci. Depuis cette page il est possible d'accéder à la page artiste de chacun des participants.
- L'écran de détails d'un artiste qui est accessible depuis chacun des trois écrans précédents. Cet écran permet notamment à l'utilisateur d'obtenir une description de celui-ci.

L'en-tête de l'application est composé d'un bouton retour permettant de revenir à la page précédente ainsi que d'un bouton permettant de revenir à la page d'accueil. La figure ci-dessous indique un exemple type de navigation pour une recherche

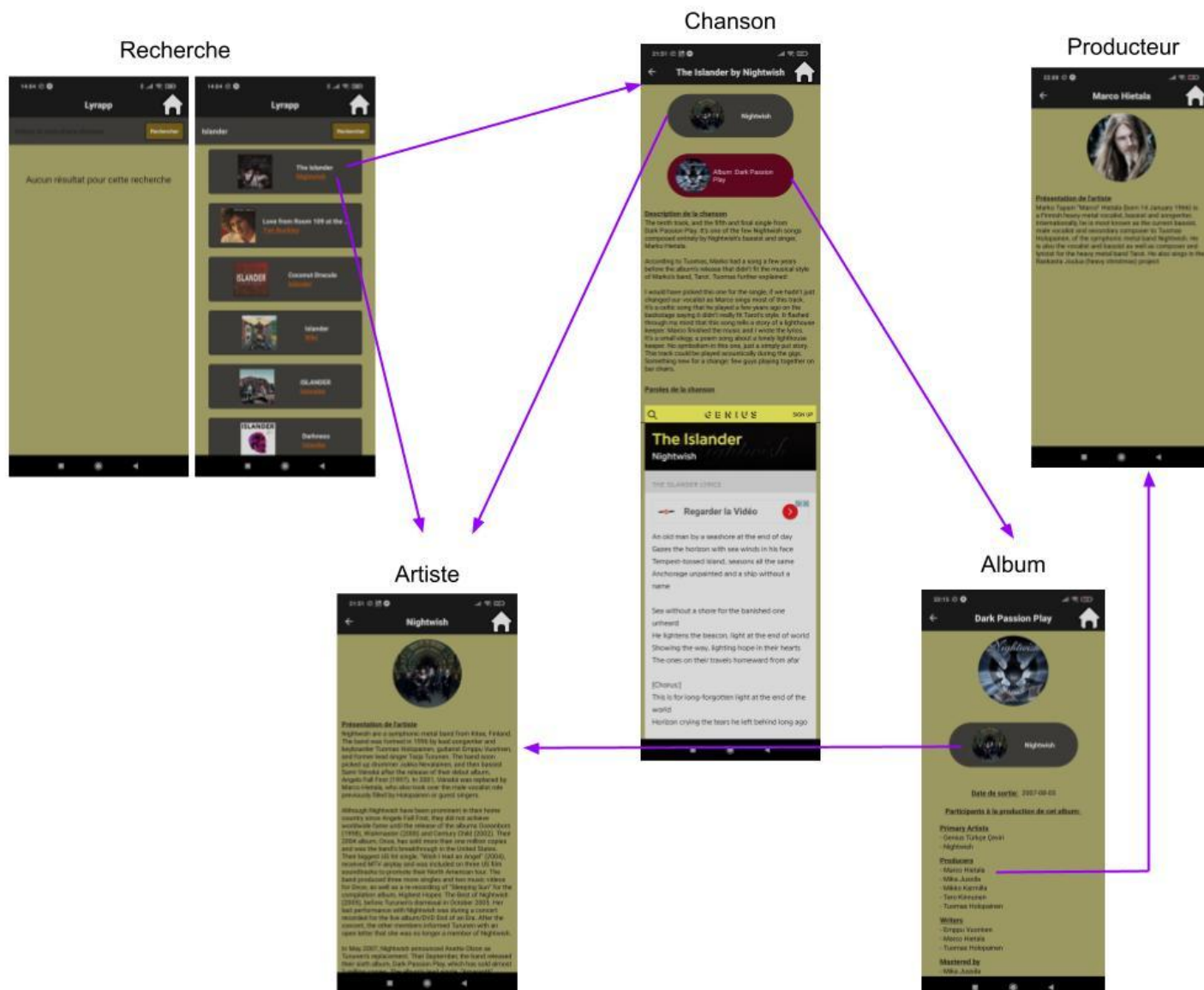


Figure 1: Diagramme de navigation

III/ Organisation

Après avoir choisi l'API nous avons réalisé tous ensemble une phase de découverte de celle-ci. Nous avons essayé différents exemples de requêtes sur Postman afin de voir les informations retournées et choisir lesquelles nous souhaitons garder. Nous avons ensuite réfléchi à l'organisation de notre application à travers la réalisation d'une maquette de celle-ci. Pour la réalisation du code nous avons opté pour la répartition suivante :

- Julie: réalisation des composants
- Charles: réalisation des écrans et de la navigation
- Loïc: réalisation des fonctions permettant le requêtage de l'API

Enfin nous avons réalisé les finitions de l'application ensemble.

IV/ Conclusion

L'application que nous avons réalisée avec l'API Genius fonctionne correctement et la communication entre l'application et l'API est fonctionnelle. Différentes pistes d'amélioration pourraient être envisagées, notamment en utilisant les annotations présentes dans l'API. Cependant, les possibilités permises par l'API restent restreintes, ne nous permettant par exemple pas d'afficher directement les paroles des chansons.