

Design

Numéro d'équipe : 17

Membres de l'équipe :

<u>Nom</u>	<u>IDUL</u>	<u>Matricule</u>
Joice Kariya	JOKAR7	111023195
Michaël Lévesque-Dion	MILED28	111090297
Samuel Morneau-Gamache	SAMOG	111099725
Bruce Perry	BRPER24	111192587
David Poisson	DAPOI65	905302625

Lancer l'application :

1. npm install
2. npm run livrable
3. Une fenêtre dans Chrome devrait s'ouvrir à l'adresse suivante:
<http://localhost:8080/#/>

Comment voir les pages :

La fonctionnalité recherche est la meilleure façon d'accéder aux différentes pages. Dans les différents résultats, le texte en vert est cliquable et mène à une page dans UBeat. Les urls sont décrits en détail dans le fichier "router/index.js".

Fonctionnalités avancées :

1. Notre première fonctionnalité est "Obtenir davantage d'informations sur un artiste". Sur la page d'artiste, nous affichons une image et une biographie. Ces données proviennent du backend de LastFM. Pour nous amuser un peu et rendre l'application plus intéressante, nous avons aussi ajouté des suggestions d'artistes similaires (dans le bas de la page de l'artiste). Sur la page d'accueil, nous avons ajouté les "top artists" présentement au Canada. Ces informations viennent aussi du backend de LastFM. Il est possible de cliquer sur l'image d'un artiste similaire ou d'un "top artist" afin d'accéder à la page de cet artiste dans UBeat. Si un artiste existe dans LastFM, mais pas dans iTunes, le lien pointe plutôt vers LastFM.

2. Pour notre deuxième fonctionnalité, nous avons modifié légèrement un endpoint du backend de UBeat et nous l'avons déployé sur Heroku : <https://fierce-tundra-31347.herokuapp.com/>. Cette fonctionnalité a été approuvée par les deux enseignants. Voici la modification qui a été faite au backend :

```
213     exports.getPlaylists = async function (req, res) {
214 +   if (!req.query.userId) {
215 +     res.status(400).send("Please provide userId query param");
216 +     return;
217 +   }
218     try {
219 -     const playlists = await Playlist.find({"owner.id": req.body.userId})
219 +     const playlists = await Playlist.find({"owner.id": req.query.userId})
220     res.status(200).send(playlists || [])
```

Comment faire pour :

Faire une recherche spécifique :

Le type de recherche est identifié par l'icône à droite de l'input "Search ..." de la barre de navigation. En cliquant sur le bouton, un dropdown apparaît qui permet de choisir le type de recherche désirée. La valeur par défaut est "Global", ce qui correspond à l'icône de loupe. Seules les valeurs autres que celle présentement sélectionnée sont affichées dans le dropdown, afin de ne pas mélanger l'utilisateur.

Supprimer une playlist ou une chanson d'une playlist :

Par rapport au dernier livrable, nous avons rendu cet aspect plus "user friendly" en diminuant le risque qu'un utilisateur supprime quelque chose par accident. Pour supprimer une playlist ou une chanson, il faut cliquer deux fois sur le bouton rouge avec la poubelle. Le premier clic fait changer la poubelle en point d'interrogation, et le deuxième clic confirme la suppression en tant que tel. Si l'utilisateur appuie une seule fois, le bouton revient à son état normal après 2 secondes.

Changer le nom d'une playlist :

1. Cliquer sur le bouton crayon blanc sur fond vert. Le nom de la playlist devient un "input".
2. Éditer le texte.
3. Appuyer sur la touche "Enter" pour confirmer la modification. Sinon, appuyer sur le bouton jaune pour annuler.

Ajouter des chansons dans une liste de lecture :

- À partir de la page d'album, il est possible d'ajouter un album complet à une liste avec le bouton "+ Album". Il est possible d'ajouter une chanson à la fois avec les boutons "+" sur chaque chanson.
- À partir de la page playlist, il est possible d'ajouter une chanson à une playlist à partir du bouton "+" sur chaque chanson. Il doit donc y avoir des chansons dans une playlist afin de pouvoir faire cette action.

Autres particularités :

Le formulaire de Login/Signup effectue une validation de base sur les champs. Le nom d'utilisateur ne peut pas être vide, le email doit avoir (grossièrement) la forme d'un email, et le mot de passe doit être entre 8 et 30 caractères. Il n'est pas possible de "soumettre le formulaire" tant que ces critères ne sont pas rencontrés.

Les noms des tracks dans les playlists sur la page d'un utilisateur peuvent ne pas avoir de liens vers la page de l'album et/ou la page de l'artiste. En effet, puisque le backend n'applique pas de restrictions au contenu d'une "track" et qu'il les sauvegarde telles qu'elles sont envoyées, il est possible que certains champs soient manquants, dont l'id de l'album et l'id de l'artiste.

Lorsqu'on est sur la page de l'utilisateur courant et qu'on "unfollow" un utilisateur, celui-ci reste dans la liste des utilisateurs "suivis", même s'il n'est techniquement plus suivi par l'utilisateur. Ceci permet à l'utilisateur de changer d'avis ou de corriger son action s'il a cliqué sur "unfollow" accidentellement. Le "count" des "following" reflète toujours le nombre d'utilisateurs suivis, et pas nécessairement le nombre d'utilisateurs dans la liste.