



JukeBoxer

An ARMV3 Presentation



Problem

- Switching music streaming services can be frustrating
- Users want to access to song data metrics when creating playlists
- Streaming services obfuscate user listening habits and song datametrics



Objective

- Create a way to transfer playlists between streaming platforms
- Show listening analytics to users
- Improve playlist creation
- Make Playlists Collaborative



Use Cases

- User wants to import data from a playlist on a streaming service into JukeBoxer
- User wants to export data from playlist on JukeBoxer onto a playlist on a streaming service
- User wants to use the data we aggregate to construct a better playlist
- User wants to collaborate and build playlist with friend on JukeBoxer



Impact

- This is a cross platform playlist creator and curation tool while other platforms focus on one platform and curation.
- This is for users who use a variety of streaming platforms.
- Allows users to collect interesting data about their music streaming activities.

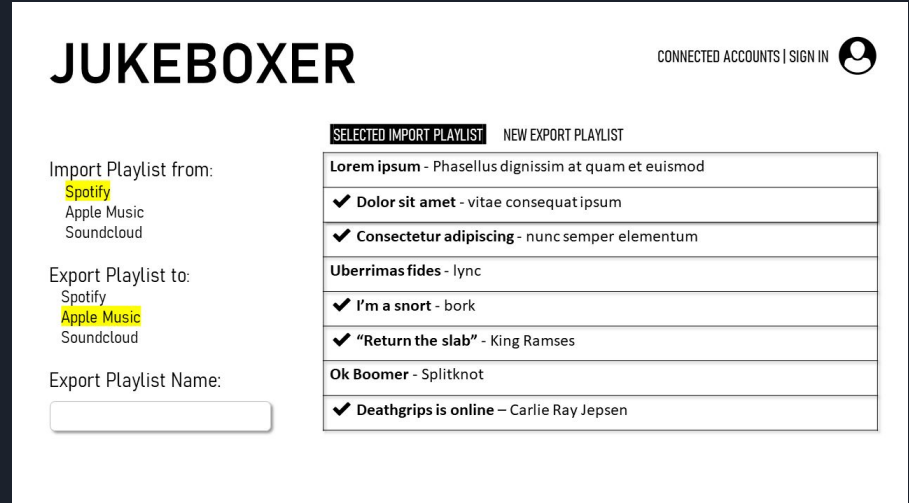


Related Work

- <https://soundiiz.com>
 - Very good platform, but doesn't have any user interaction or music analytics. Syncs playlists, artists, tracks, and albums across most major streaming platforms.
- <https://playlistconverter.net/>
 - Allows sharing across most popular music streaming platforms, but does not have a wide range of services and no user analytics.

Design (Frontend)

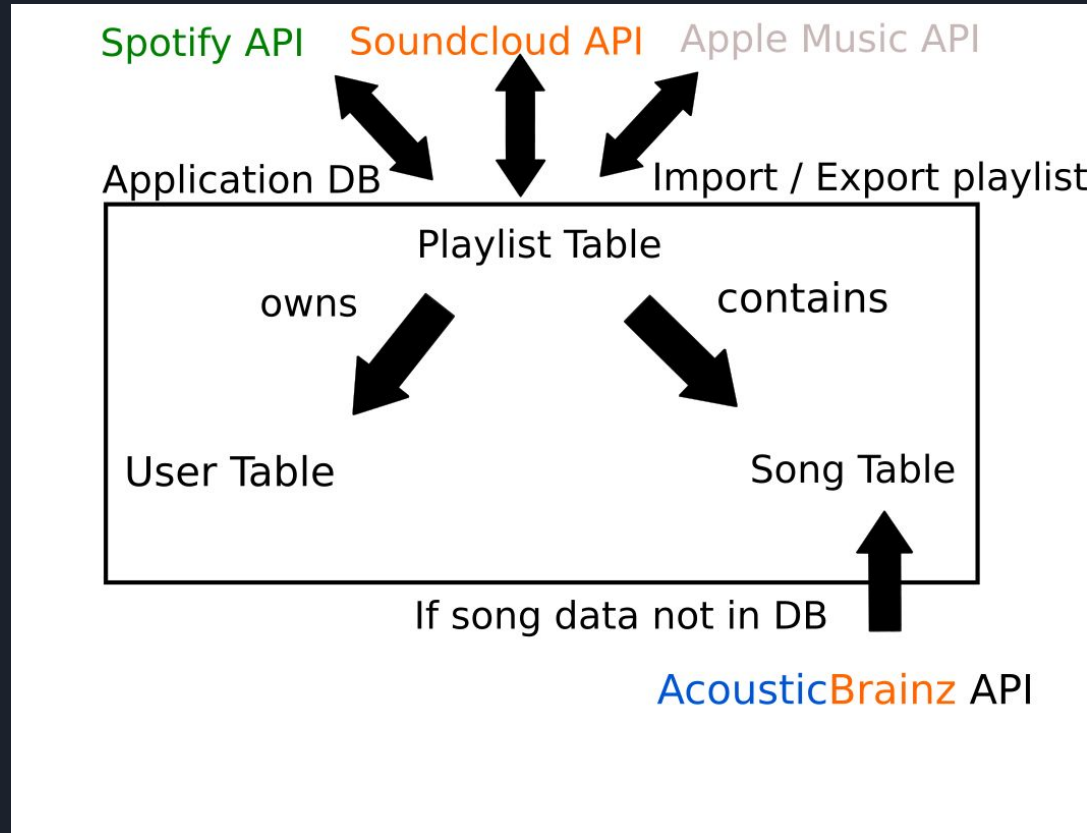
- Node.js web app utilizing OAuth for authentication
- User will authenticate with any of the supported streaming platforms
- Chosen playlists will be extracted and presented for editing
- User can then export the playlist to the platform of their choice



The screenshot displays the 'JUKEBOXER' web application interface. At the top right, there are links for 'CONNECTED ACCOUNTS' and 'SIGN IN' with a user profile icon. The main content area is divided into two sections: 'SELECTED IMPORT PLAYLIST' and 'NEW EXPORT PLAYLIST'. Under 'SELECTED IMPORT PLAYLIST', there are three options: 'Spotify', 'Apple Music', and 'Soundcloud'. Under 'NEW EXPORT PLAYLIST', there are three options: 'Spotify', 'Apple Music', and 'Soundcloud'. Below these options, there is a text input field for 'Export Playlist Name:'. To the right, a table lists several playlists with checkboxes for selection. The table has two columns: 'SELECTED IMPORT PLAYLIST' and 'NEW EXPORT PLAYLIST'. The first column contains the following items: 'Lorem ipsum - Phasellus dignissim at quam et euismod', 'Dolor sit amet - vitae consequat ipsum', 'Consectetur adipiscing - nunc semper elementum', 'Uberrimas fides - lync', 'I'm a snort - bork', 'Return the slab' - King Ramses, 'Ok Boomer - Splitknot', and 'Deathgrips is online - Carlie Ray Jepsen'. The second column contains checkboxes, with the first three (Dolor sit amet, Consectetur adipiscing, and I'm a snort) checked.

SELECTED IMPORT PLAYLIST	NEW EXPORT PLAYLIST
	<input type="checkbox"/> Lorem ipsum - Phasellus dignissim at quam et euismod
<input checked="" type="checkbox"/>	<input type="checkbox"/> Dolor sit amet - vitae consequat ipsum
<input checked="" type="checkbox"/>	<input type="checkbox"/> Consectetur adipiscing - nunc semper elementum
	<input type="checkbox"/> Uberrimas fides - lync
<input checked="" type="checkbox"/>	<input type="checkbox"/> I'm a snort - bork
<input checked="" type="checkbox"/>	<input type="checkbox"/> "Return the slab" - King Ramses
	<input type="checkbox"/> Ok Boomer - Splitknot
<input checked="" type="checkbox"/>	<input type="checkbox"/> Deathgrips is online - Carlie Ray Jepsen

Design (Backend)





Tasks

1. Gain **background knowledge** of all service provider APIs.
2. Design the **basic implementation** of what API calls we will need to make, as well as how they will feed into our website.
3. Get basic **server architecture** and **database** structure running.
4. Program the **home page** of our application.
5. Make the **web page** for **playlist import** and **export**.
6. **Connect** to each of the streaming services via OAuth
7. **Import playlists** from each of the streaming services via their API
8. **Export playlists** on the website to each of the streaming services via their API
9. Pull information from **AcousticBrainz** about tracks in playlists
10. **If time allows**, create functionality to allow for the **following and creation of a friends list**.



References

Serra, Xavier. "Welcome to AcousticBrainz!" AcousticBrainz, 31 July 2019, acousticbrainz.org/.

Soundiiz. "Transfer Playlists and Favorites between Streaming Services." Soundiiz, soundiiz.com/.

"Playlist Converter - Convert & Share Playlists and Transfer Them between Spotify, Napster, Deezer." [Http://Playlistconverter.net/](http://Playlistconverter.net/), playlistconverter.net/.