**Feature Request: Progress/Status Bar**

July 3, 2019

**OBJECTIVE**

The goal of this feature request is to provide an obvious confirmation that the playlist has been successfully made, or if an error occurs, let the user know why.

**BACKGROUND**

Currently, the base Jamming project only refreshes the Playlist tracks and the playlist name as confirmation that *something* has happened. As I discovered through development, this doesn’t always mean that the playlist was created: sometimes the playlist was created, but the name wasn’t kept. Others, the information just refreshed, but the playlist was nowhere to be seen. And in a world of trophies and participation ribbons, this simply will not do!

This feature will do the following upon completion:

* Add multiple values for a ‘status’ state.
* As the user progresses through the process of adding songs and naming the playlist, a status bar will begin to fill.
* Additionally, as progress is made, encouraging, music-themed messages will display, letting the user know that their changes are successful.
* If any errors pop up, the progress bar and progress messages will let them know what is happening.

**TECHNICAL DESIGN**

**StatusBar**

Along with the existing components, a new component will be created, called **StatusBar**. This component will be placed beneath the search bar and take roughly 15-20vh. On page load, the user will only see an ‘empty’ darker colored bar. The component will have a **status** attribute that receives a number (from 0 to 100, defaultValue={0}) from the parent component.

Styling for Status bar will consist of two sections, a top and bottom. The top section will consist of the status bar. This will be two overlapping divs, with classNames of ‘background’ and ‘status-level’. As the names suggest, ‘background’ will be the darker bar in the back while ‘status-level’ will be on top showing the user their progress. As the **status** attribute is updated, JavaScript will update the width of ‘status-level’ giving the appearance of the progress bar filling up.

The bottom section of this component will be where any status messages or instructions will be listed. While the ‘status’ remains in its default state, instructions will appear to click the search button or start typing a search term. As soon as they have begun typing or have logged in, the progress bar will start to fill, and the instructions will change accordingly, displaying encouraging messages.

**checkStatus(errStatus)**

In the App component, a new method will be created, called **checkStatus()**. This **checkStatus()** method will be added as an **onUpdate**attribute *to* the **SearchBar** and **Playlist** components. This prop will then get added as event handlers to both the **SearchBar**’s input and the **Playlist’s** playlist name **<input>**, and **Tracklist**. Once the user has logged in, a search term is provided, playlist name given, or track added to the **Playlist** <**TrackList/>**, the **checkStatus()** method will begin adding up the progress. Each item will be given equal priority, adding an equal percentage to a *percentage* variable.

Once the *percentage* variable is greater than zero, the ‘status’ state will be updated to ‘Almost There…’ to signify that the process has begun. Once the *percentage* variable equals 100, the state will be changed to ‘Jamming’, which will display a new message to the tune of “Click Save Playlist to being Jamming!” underneath the **SearchBar** in the new **StatusBar** component.

**errStatus (state)**

In the event that something does go wrong, the errStatus state will change the coloring of the status bar. The default value will be ‘working’, changing the progress bar purple. Once *percentage*={100} and the playlist successfully saves, errStatus will be set to “no errors” and the bar will turn green. If, however, the Spotify API doesn’t connect, an accessToken is not retrieved, or a playlist isn’t created, errStatus will be set to “errored” and an error code will be concatoneated to the end, depending on where the error occurred. If this happens, errStatus will be sent to checkStatus to have the error code processed and display the correct message.

**Progress Rocks!**

**Jammming!**

**Rap! Something isn’t right!**

**Try that one more time…**

**CAVEATS**

**Request Errors**

A potential issue with the proposed issue is that if there’s a request error, there isn’t always an easy fix and sometimes the best thing to do would be refresh the page. This would cause the user to lose all progress, which would make no one happy. However, the error codes for the most part will be set throughout the workflow (connecting to spotify, retrieving song lists, adding songs, changing playlist name, etc.) and as such, could have a defined set of instructions to fix the issue.