Jammming Discover Tool

**January 20th 2018**

**OBJECTIVE**

I’d be interested in adding a Discover New Artist feature on the Jammming Application. Sometimes, a listener wants to vibe to a fresh face and break the monotony.

**BACKGROUND**

The Discover New Artist Tool will be a new Search button that allows a listener to discover a new artist based on their interests and listening habits. This would be a great way to not only keep listeners interested in the app but also expose brand new artists.

This new feature should perform the following actions:

* Display a current list of “recently played” and “top played” songs / artists
* Perform a search for an artist that is like an artist within the playlist
* Select the most popular song from a new artist determined by the search

**TECHNICAL DESIGN**

**Retrieve “Recently Played” or “Top Played”**

A new component, discoverNewArtist should be created. This component should be designed to receive all the previous artists and playlists that the listener has interacted with the most. The component discoverNewArtist should contain a key of artists and songs from the listener that are among the most played.

To receive this information, a new method, **Jammming.retrieveMostPlayed()**, that hits the endpoint. The endpoint requires the user ID. We can refractor this request into a new method called **Jammming.getCurrentUserId()** to avoid duplicate code. At the top of App.js, we will instantiate a variable called ‘userId’ with no value. Then inside **Jammming.getCurrentUserId()**, we will check to see if userId's value is already set (from a previous call to the function). If it is, we will create and return a promise that will resolve to that value. Otherwise we will make the call to the /me endpoint and return that request's promise.

Once **Jammming.getCurrentUserId()** is written, we should use it in both **Jammming.savePlaylist()** and our new method, **Jammming.retrieveMostPlayed()**.

In **Jammming.retrieveMostPlayed()**, once the user ID has been retrieved, we can make our call to the /users/{user\_id}/tracks endpoint. Upon completion of this request, we will update the current track's state to an array of the returned playlists. Rather than storing the entire playlists of related tracks, we should store a track that contains a relation to the name of each artist in the user’s most played.

To call this method at time of render, we should add a .componentWillMount() lifecycle method to newArtist and call .retrieveMostPlayed within it.

The .render() method of newArtist should render a list of newArtistListItem components (another new component), passing down the ID and name of the related track to be rendered.

Lastly, the App should render newArtist.

**CAVEATS**

**Tracks that have already been heard**

The new tool will also eliminate tracks found in the user’s recently played library. An array linked to the user’s Recently Played list will be wired to skip any tracks already present in the user’s library regardless if they are related or not. This should be done by making an empty array with quotes linked to the Recently Played section. Any song found there will be skipped.