Feature Request: Show and Update Existing PlayLists

OBJECTIVE

To give users the ability to see existing playlists and show in our App and allow them to delete tracks from existing playlists.

BACKGROUND

Currently Jamming supports the ability to search songs and create playList by adding songs. But it does not show existing Playlists and songs underneath existing playlists. This feature will accomplish the following:

- It will display a list of current user's playlists in Jamming Project
- It will display songs underneath existing playlists
- It will allow users to delete tracks from existing playlists

TECHNICAL DESIGN

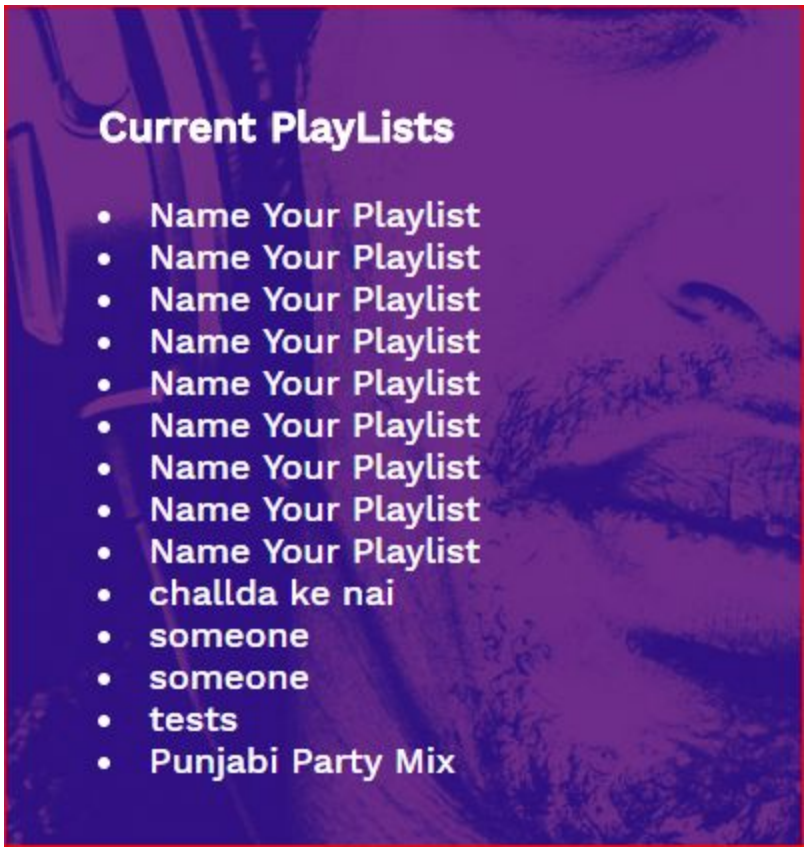
Display Existing PlayLists

1. To display existing playlists, we have to create a new Component - ListofPlayLists. It will retrieve and display current user's playlists.
2. Then in Spotify, we need to create another function- Spotify.getCurrentUsersPlaylists() that will call the following endpoint: https://api.spotify.com/v1/users/{user_id}/playlists
   Here is reference from Spotify (https://beta.developer.spotify.com/documentation/web-api/reference/playlists/get-list-users-playlists/). In this endpoint we need user_id, which we can get in similar way we are getting in savePlayList method of Spotify.
    1. Once the User ID is retrieved we will call the endpoint listed above to provide us list of playlist's for the current user.
    2. We will use this result in our component ListofPlayLists to display those playlists.

```
getListOfPlayLists() {
    console.log(`User has following playList`);
    const playlist = Spotify.getCurrentUsersPlayLists().then(results => {
       this.setState({listOfPlayLists: results});
            });
 }
```

3. Since we want to display these playlists on render- we will add above function in componentWillMount() function.
4. Then, we need to call this component inside App to render it

Finally after App will render it, and the component will look like following :



Select PlayList

We need to allow user to click on the playlist to retrieve the entire playlist.

PlayListItem- new component should be created. This component will retrieve the selected user's playlist. Inside ListofPlaylists - we will initialize state of selectedPlaylist to empty element. To retrieve selected playlist - we will have two functions- one function to set selectedPlaylist once we click/select the playlist.

```
showPlaylistOnClick(playListID,playListName) {
  console.log("show Playlist playlist");
  this.setState({selectedPlayList:{id:playListID,name:playListName}});
  console.log(this.state.selectedPlayList);
}
```

We will invoke this method where we are showing playlists in ListofPlaylists as following:

```
<h3>Current PlayLists</h3>
{this.state.listOfPlayLists.map((playlist) => <li key={playlist.id}><a
onClick={this.showPlaylist}>
  <a onClick={(e) => this.showPlaylistOnClick(playlist.id, playlist.name)}>
  {playlist.name}</a> </a>
</li>)}
```

And another function to retrieve selected playlist and show using component PlayListItem. To retrieve selected playlist, we will need another function- getPlaylist(Id) in Spotify.js to return all the tracks inside the playlist using following endpoint: https://api.spotify.com/v1/users/{user_id}/playlists/{playlist_id}/tracks

And we will pass in selectedPlaylist property to this function getPlaylist(selectedPlaylist) which will return us selected playlist and we will pass in this to PlayListItem to show selected Playlist.

Delete Tracks from the Playlist :

After we show tracks using PlaylistItem - we will allow user to delete tracks from the existing playlist- we will add buttons beside tracks which will serve the functionality of deleting and we will add another function in Spotify to delete the track using following endpoint:

For this, first we need to get trackURI - following endpoint will return trackUris of all.

/v1/users/{user_id}/playlists/{playlist_id}/tracks

First- we need to store trackUri of specific track we are going to delete and then call the following endpoint using Spotify.js

DELETE

/v1/users/{user_id}/playlists/{playlist_id}/tracks

CAVEATS

Structure of the new Components

With current implementation- I have been storing states differently in different components. The structure can be refined better to make it more clear with proper readability- Passing information between different components can be refined further. Code can be more clean so any one can understand it easily.

Asynchronous Requests

Since I have been storing states differently in different components. It could cause problem if same user is logged into multiple devices and trying to update playlists at the same time.