Feature Request: PreviewPopUp (PPU)

**February 24th 2019**

# **OBJECTIVE**

The feature previews chosen track in a popup box with information about the album, artist, and track length before adding the track to the playlist. The tracks may be added to the playlist from the preview box or the search result list.

# **BACKGROUND**

The implementation of the feature makes it easier to choose tracks for adding to a new playlist. It fetches additional information of the tracks that the original Jammming App are lack of.

Because it could be difficult to determine the tracks’ version from the original Jammming App’s functionality, PreviewPopUp feature displays from which album, album type and release date of that album the track belongs to. As a bonus, the album’s cover image is displayed as a visual effect.

You can see the chosen track’s length and track number in current album and may as well add the track directly to the playlist from within the PreviewPopUp box.

The tracks in the search result’s list should be assigned a link that opens the feature box on clicking. The box has the functionality of adding the tracks to the playlist, or may be closed without action.

# 

# **TECHNICAL DESIGN**

All data are fetched from the same (track) object in the Spotify Object Model that are already set up in the **Spotify.js** file. You just have to add the following API keys to the returned track object:

* *duration\_ms,*
* *track\_number,*

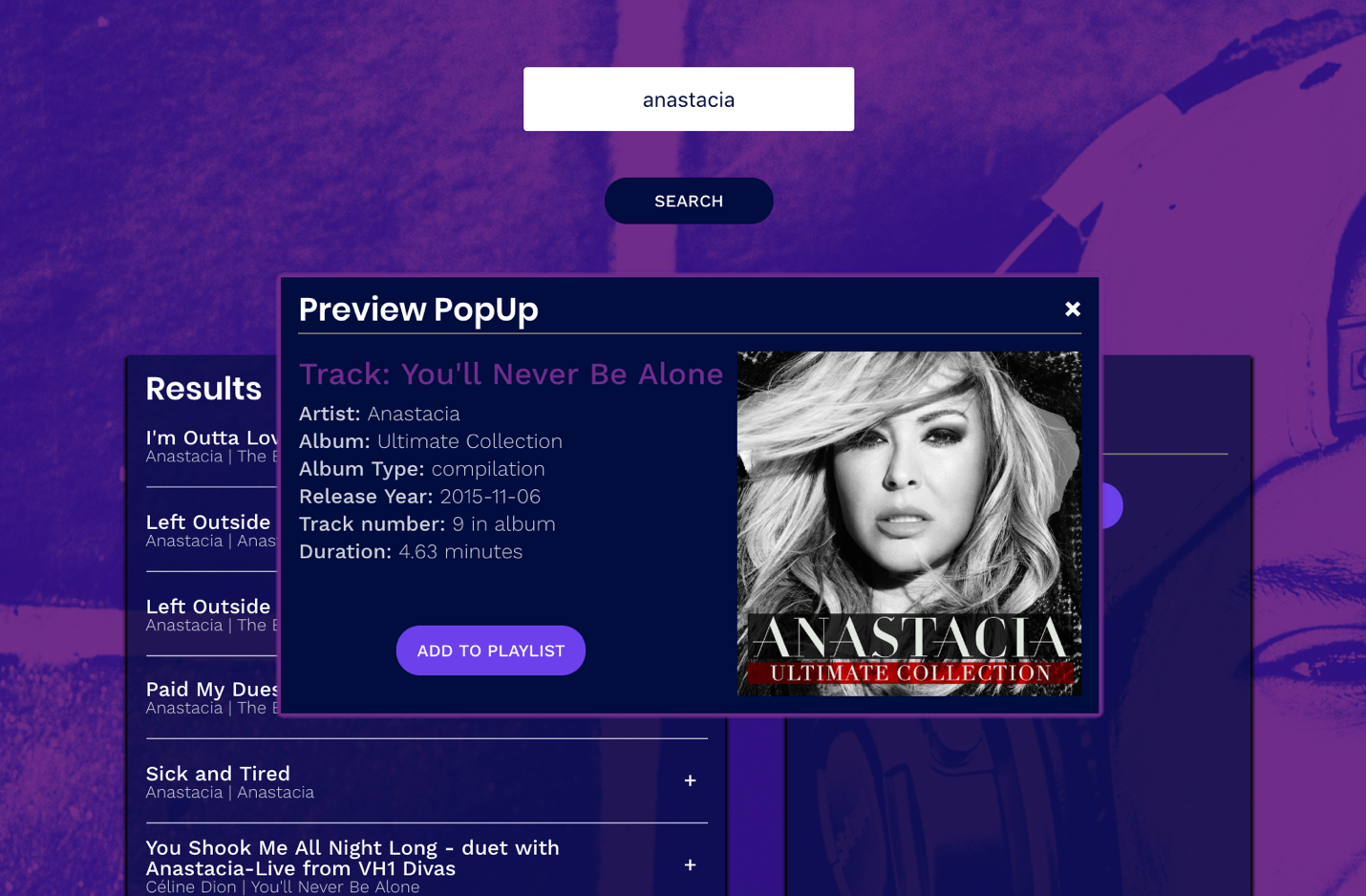
and these to the inner album object:

* *album.album\_type,*
* *album.images[1].url,*
* *album.release\_date.*

**Duration\_ms** (track length in milliseconds) prop value has to be converted to minutes.

**Album.images[x].url** contains three sets of images in different sizes, middle-size (index 1 and 300px times 300px) are good enough for this project.

To get a clue of how the implementation of the feature component should look like and work, a mockup is included below:



## Implementation Walkthrough

1. Create a **PreviewPopUp** component class in a PreviewPopUp directory inside src/components directory where all the other components are located.  
     
   The component should render HTML for chosen track’s selected attributes (name, artist, album, album\_type, release\_date, track\_nr, length and cover\_album).  
     
   You can reuse method addTrack that are passed through the app. Create a method for closing the PreviewPopUp div on click event.
2. Create a **previewTrack state** array in app.js to hold the current previewTrack and set state in a function **previewTrack** to pass to the PreviewPopUp Component.
3. Put the PPU component in the App component and pass the previewTrack state and addTrack method as attributes onPreview and onAdd.
4. Pass the previewTrack function through the SearchResults, TrackList ending up in the Track component. Use vanilla javascript to display the PPU div (originally display:none in css) in the function.   
     
   The preview tracks can be render through an if statement regarding on the isRemoval prop. Make the rendered track name a clickable link where isRemoval is false, i.e. the search results list.   
     
   The link should execute the previewTrack function, that opens (display:block) the preview box and render the preview tracks HTML.
5. Use the addTrack method to save the track from the preview box to the play list.

# **CAVEATS**

You might choose to put the PPU in the Track component, using the passed props. In that case be aware that all tracks may be rendered as unique preview windows.