Feature Request: Full keyboard control

**August 13th 2018**

# **OBJECTIVE**

Power user getting things done within an app by mostly keyboard keys. Keyboard use leads to a faster interaction within an app and saves precious time. Jammming has to focus on this user group and should add keyboard combos for intuitive interaction with different features like “start a new search with CTRL + F” or “save a playlist with CTRL + S”.

# **BACKGROUND**

In most software applications keyboard combos are a basic feature. The majority of the user is familiar with the CTRL+ALT+DEL combo to start the windows task manager and end (kill) a task that does not respond, CTRL+TAB to switch between applications, or CTRL+N in the windows explorer to create a new folder. The list goes on, but the main point is this: Keyboard combos are shortcuts for getting things done with less time than with a mouse.

Those combos are intuitive (the N key in CTRL+N can be associated with “new”). Through implementing keyboard combos in jammming, the user will be able to create a playlist in less time than he has with a keyboard and mouse combination. He does not need to get his fingers off the keyboard.

The following features should be implemented:

* A new search will be triggered by the key combo **CTRL + F** (the prior search request will be removed from the searchlist)
* **Enter** will initialize the search with the user input
* The first track in the searchlist (list on the left side with the results) should be highlighted and ready for more user interaction
* Inside of the searchlist, the user can navigate with the arrow keys
* With a press on the **right arrow** key, the current track will be transferred to the playlist (list on the right side)
* If the user want to customize her playlist, she can press **CTRL + TAB** to toggle between the searchlist and playlist
* Afterwards she can modify her playlist with the **left arrow** key to remove tracks
* By pressing **CTRL + S**, the playlist will be saved at the given Spotify account
* If a new playlist should be generated, **CTRL + N** starts a new one

# **TECHNICAL DESIGN**

The following existing components will be updated:

## **App.js**

Gets an “componentDidMount” lifecycle (see this [link](https://www.andreasreiterer.at/reactjs-lifecycle-methods/) for more information about reacts lifecycle) according to the following example:

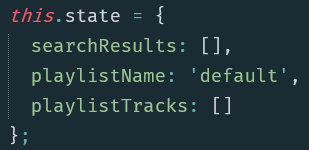


The function “handleKeyEvent” should have the following if/else statements (please note: this is not the final version and can be updated or modified by the given developer or system architect):



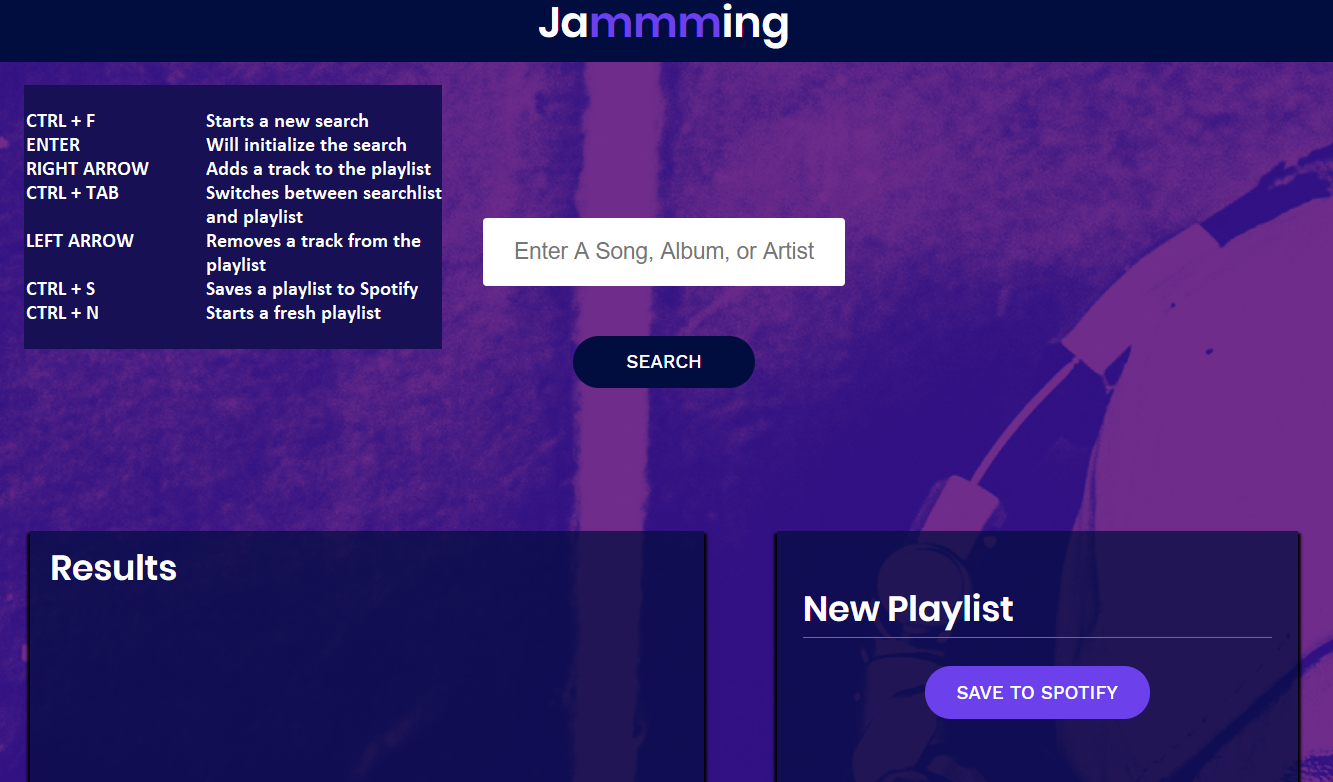
Most functions are implemented, but the following two need to be added:

* *newPlaylist* = sets the state of App.js to the default values:



* *toggleSearchlistPlaylist* = Toggles the highlighting between the searchlist and the playlist, see highlighting of the track component in section TrackList.js / TrackList.css.

For user experience reasons a tooltip should be added. It should be placed clearly visible at the top left or right corner. The following is a draft how it might could look like (same background color as “Searchlist” or “Playlist” preferred) :



## **SearchBar.js**

The HTML element <input> should get an “onKeyPress” event, that triggers the “this.search” method. The search method itself should do its function only if the enter key (keyCode 13) or the button was pressed by mouse.

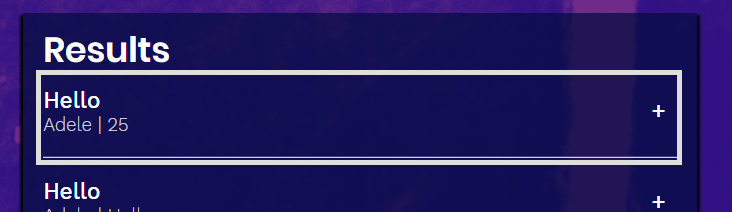
## **Playlist.js**

Implementation of the keypress function is similar the enter key in the SearchBar.js component:

The HTML element <a> should get an “onKeyPress” event that triggers the “this.props.onSave” method, whose origin is in the App.js component. There, the “savePlaylist” method should only save the playlist to Spotify if CTRL and S (keyCode 83) was pressed. See the “handleKeyDown” method in section App.js for further information.

## **TrackList.js / TrackList.css**

Gets a new CSS class to display the highlighted track component. The following image is an example of the highlighting, but should not be treated as final version:



# **CAVEATS**

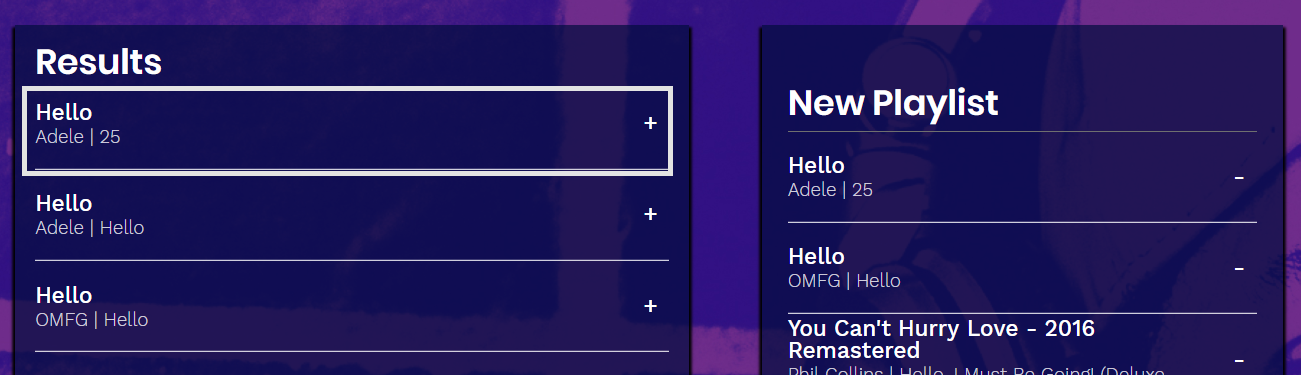
## **App.js: Potential drawbacks**

The following two methods need attention:

* addTrack
* removeTrack

They should only work on the highlighted track in the searchlist or playlist.

In the following example, the first track in the searchlist (left side) is highlighted and only this track should be added to the playlist (right side) through the “addTrack” method:



The opposite applies to the “removeTrack” method with the given key (left arrow key).

### **Alternative solution**

This feature may be canceled due to customer feedback and needs more investigation. It may be easier to navigate and customize both lists via mouse than with keyboard.

If this feature will be canceled, the “toggleSearchlistPlaylist”method and the CSS class in TrackList.css will not be needed and can be skipped.

## **SearchBar.js: Potential drawbacks**

Excessive use of the enter key should not be supported: If the input contains the same string, a request to the Spotify API should not be performed and the searchlist will not update.