

# Jammming Feature: Searching when 'Enter' is pressed

February 4<sup>th</sup> 2018

## OBJECTIVE

To give users the ability to activate the Search button when pressing the 'Enter' key.

## BACKGROUND

Currently, Jammming supports the ability of jump starting the search mechanism for tracks to add them to personalized playlist only by clicking the 'Search' button. However, it does not support activating this mechanism by pressing the 'Enter' key. With people having a automatic response of clicking Enter every time they want to submit something, this is an essential feature to implement. This feature allows the user to activate the search by pressing Enter.

## TECHNICAL DESIGN

### *Adding the Enter key as a way to search*

Inside the **SearchBar.js** file in the **Components** directory, a new method called ***handleKeyDown()*** should be created. This method will be the event handler to the event listener *onKeyDown*.

***handleKeyDown()*** should accept *event* as a parameter. On the body of ***handleKeyDown()*** there should be a conditional statement that checks if the property **.key** of *event* is equal to 'Enter'. If that is the case, then the body of the conditional statement should call *this.props.onSearch* with *this.state.searchTerm* as its parameter.

After that, on the input JSX element in the **render()** method of **SearchBar.js**, the **onKeyDown** event listener should be set as a property to the value of **this.handleKeyDown** (and don't forget the curly braces to make sure the code know this is written in JavaScript).

Make sure the **redirectURI** constant in the **Spotify.js** is set to the one you intend it to be redirected to.

## CAVEATS

There are three options that could be used for Event Listeners: **onKeyUp**, which is fired when the key on the keyboard you are pressing is released, **onKeyDown**, the one I chose, which is fired when you press down a key on the keyboard, and **onKeyPress**, which is fired when you press down a key that also represents a character (letter, number, etc) on the keyboard.

The choice of **onKeyDown** was made on the premise that **onKeyPress** would require a **KeyCode**, making the code more complicated to be written and understood, and also because it would also have the extra step to evaluate the pressed key as a character or not. As for **onKeyUp**, **onKeyDown** just seemed like the simplest between the two.

### **onKeyPress**

In case the user wants to use the event listener **onKeyPress**, the conditional statement in the event handler's body should check the event **.keyCode** property instead of just **.key**, and is should check if this property is equal to **13**, the keyboard code for the Enter key. Additionally, the event listener should be changed to **onKeyPress**.

### **onKeyUp**

In case the user wants to use the **onKeyUp** event listener, the implementation is the same as for **onKeyDown**, but the event listener should be **onKeyUp** on the **render()** method.