

Feature Request: Enter Triggers Search

June 05 2018

OBJECTIVE

To allow the user to trigger a search in the search bar by pressing enter on the keyboard.

BACKGROUND

Currently, Jammming will only trigger a search in its database if the search button is pressed manually and at the moment if the enter key is pressed, nothing will happen until code that supports this feature is implemented.

This feature will allow the user of Jammming to search the Jammming database, once information has been inputted, by pressing the enter key. This should have the same effect as pressing the search button manually. This will add convenience and a user friendly interface feature to the Jammming app.

TECHNICAL DESIGN

The goal of this technical document is to outline the method implemented to allow for the enter key to trigger a search the same way clicking the search button would.

*All changes have been made to and implemented into the main **SearchBar.js** file uploaded in the previous project submission.*

The first thing needed is the java “code” for the **enter key**. With a bit of research, it is apparent that the **enter key** is represented by the number **13** in the script.

The line of code that would prompt the computer to accept the **enter key** would look as follows:

`(event.key === 13)`

A new **function** must be created with the header:

```
handleKeyPress(event) = {}
```

This will allow for the **enter key** to be accepted as an event input for the console to read.

Inside the function, an **'if' statement** must be created with the following parameters:

```
if (this.state.term) {}
```

And inside this, another **'if' statement** with the parameters of the **(event.key === 13)** created earlier.

Finally, an **argument** for the **'if' statement** must be created with the parameters:

```
this.props.searchSpotify(this.state.term)
```

This will allow the search to take effect if the **enter key** is pressed.

The entire function should look as follows:

```
handleKeyPress(event) = {  
  
  if (this.state.term) {  
  
    if (event.key === 13) {  
  
      this.props.searchSpotify(this.state.term)  
  
    }  
  
  }  
  
}
```

In order for the console to recognise the new function when the search bar is used, it must be added to the **constructor** database. An instance of the function should be created under the

main `super(props)` and `this.state` conditions, directly under **`this.handleChange = this.handleChange.bind(this);`** in the presented code.

This can be done by entering **`this.handleKeyPress = this.handleKeyPress.bind(this);`**

The final **constructor** element should look as follows:

```
constructor(props) {  
  
  super(props);  
  
  This.state = {  
  
    Term = ''  
  
  };  
  
  this.handleSearch = this.handleSearch.bind(this);  
  
  this.handleChange = this.handleChange.bind(this);  
  
  this.handleKeyPress = this.handleKeyPress.bind(this);  
  
}
```

Finally, inside the main `<div>` in the **return() function**, a line of code should be added so that the actual pressing of the key will trigger the search.

`onKeyPress = {this.handleKeyPress}`

This will allow for the Search Bar to recognise the enter key as a means of triggering the search.

The final edited `<div>` line under the **return()** function should look as follows:

```
return(  
  
  <div className="SearchBar" onKeyPress={this.handleKeyPress}>
```

CAVEATS

Accidental search triggering

Due to the nature of the code, pressing the enter key will result in a search being carried out regardless of what is inside the search bar. In addition, pressing the enter key whilst on the app may still trigger a search if the user has clicked on the search bar and then clicked off of it, as the console may not recognise the action cancellation when leaving the search bar.