To implement keyup and keydown events for the `react-native-searchable-dropdown` component, you can follow these steps:

1. Install the package:

npm install react-native-searchable-dropdown

2. Import the necessary components and modules in your file:

import React, { useState } from 'react';

import { View, TextInput, FlatList } from 'react-native';

import SearchableDropdown from 'react-native-searchable-dropdown';

3. Create a functional component and define an array of data for the dropdown options:

const MyComponent = () => {

const [data, setData] = useState([

{ id: 1, name: 'Option 1' },

{ id: 2, name: 'Option 2' },

{ id: 3, name: 'Option 3' },

]);

const [selectedItem, setSelectedItem] = useState(null);

// Function to handle keyup event

const handleKeyUp = () => {

console.log('Key up event');

// Perform desired action on keyup event

};

// Function to handle keydown event

const handleKeyDown = () => {

console.log('Key down event');

// Perform desired action on keydown event

};

return (

<View>

<SearchableDropdown

onTextChange={(text) => console.log(text)} // Handle text change event

onItemSelect={(item) => setSelectedItem(item)} // Handle item selection event

containerStyle={{ padding: 5 }}

textInputStyle={{ padding: 12 }}

itemStyle={{ padding: 10 }}

itemTextStyle={{ color: '#222' }}

itemsContainerStyle={{ maxHeight: '60%' }}

items={data}

defaultIndex={0}

placeholder="Select an option"

resetValue={false}

underlineColorAndroid="transparent"

keyUp={handleKeyUp} // Assign keyup event handler

keyDown={handleKeyDown} // Assign keydown event handler

/>

<TextInput

style={{ marginTop: 20, paddingHorizontal: 10, height: 40, borderColor: 'gray', borderWidth: 1 }}

placeholder="Type something"

onKeyUp={handleKeyUp} // Assign keyup event handler

onKeyDown={handleKeyDown} // Assign keydown event handler

/>

</View>

);

};

4. Use the `SearchableDropdown` component and provide the `keyUp` and `keyDown` event handlers as props.

The `SearchableDropdown` component also supports other props for customization. Feel free to adjust them according to your needs.

5. Additionally, you can add `onKeyUp` and `onKeyDown` event handlers to a regular `TextInput` component to handle key events.

Remember to import and render the `MyComponent` in your main application file or any other appropriate file within your React Native project.

By following these steps, you should be able to handle keyup and keydown events using the `react-native-searchable-dropdown` component and a regular `TextInput`.