Searching using Search Bar Filter in React Native List View

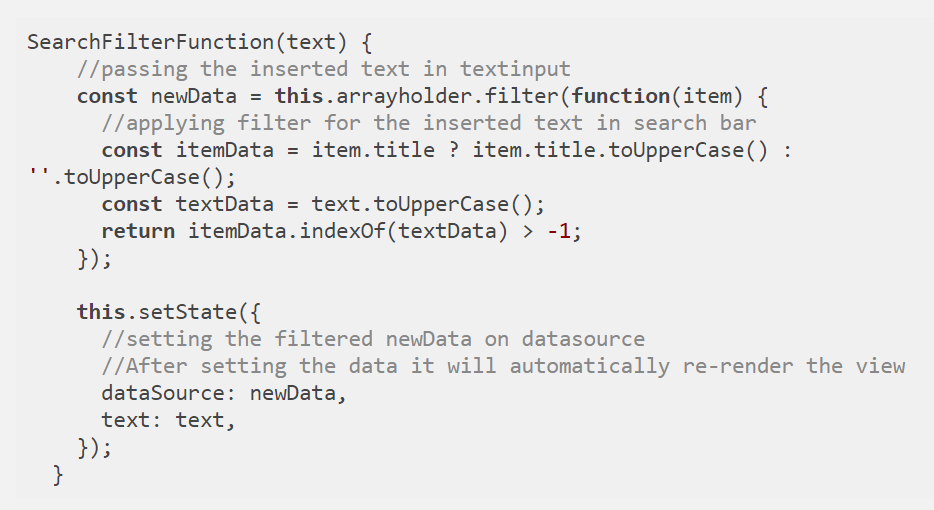
This is an example to Make Search Bar Filter for ListView Data in React Native. Basically, We will make a React Native FlatList with real-time searching ability. If we have a Long list in the app then it is very inconvenient to search the required data by scrolling the whole list.

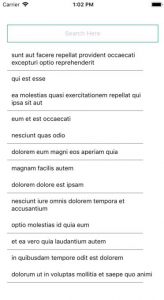
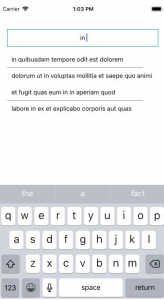
Let’s take an example of E-Commerce which has the number of products and we have to choose our required product. You can imagine how tough it will be to search for a product by scrolling. So at this point, we need a Search Bar Filter on Listview so that we can easily search the required data and can save time.

For Real-Time Searching in ListView using Search Bar Filter

The logic is simple

* We will load the list from the network call and then show it to the user.
* The user can search the data by entering the text in TextInput.
* After inserting the text SearchFilterFunction will be called
* We will compare the list data with the inserted data and will make a new Data source.
* We will update the data source attached to the ListView.
* It will re-render the list and the user will be able to see the filtered data.



//This is an example code to Add Search Bar Filter on Listview//

import React, { Component } from 'react';

//import react in our code.

import {

Text,

StyleSheet,

View,

FlatList,

TextInput,

ActivityIndicator,

Alert,

} from 'react-native';

//import all the components we are going to use.

export default class App extends Component {

constructor(props) {

super(props);

//setting default state

this.state = { isLoading: true, text: '' };

this.arrayholder = [];

}

componentDidMount() {

return fetch('https://jsonplaceholder.typicode.com/posts')

.then(response => response.json())

.then(responseJson => {

this.setState(

{

isLoading: false,

dataSource: responseJson

},

function() {

this.arrayholder = responseJson;

}

);

})

.catch(error => {

console.error(error);

});

}

SearchFilterFunction(text) {

//passing the inserted text in textinput

const newData = this.arrayholder.filter(function(item) {

//applying filter for the inserted text in search bar

const itemData = item.title ? item.title.toUpperCase() : ''.toUpperCase();

const textData = text.toUpperCase();

return itemData.indexOf(textData) > -1;

});

this.setState({

//setting the filtered newData on datasource

//After setting the data it will automatically re-render the view

dataSource: newData,

text: text,

});

}

ListViewItemSeparator = () => {

//Item sparator view

return (

<View

style={{

height: 0.3,

width: '90%',

backgroundColor: '#080808',

}}

/>

);

};

render() {

if (this.state.isLoading) {

//Loading View while data is loading

return (

<View style={{ flex: 1, paddingTop: 20 }}>

<ActivityIndicator />

</View>

);

}

return (

//ListView to show with textinput used as search bar

<View style={styles.viewStyle}>

<TextInput

style={styles.textInputStyle}

onChangeText={text => this.SearchFilterFunction(text)}

value={this.state.text}

underlineColorAndroid="transparent"

placeholder="Search Here"

/>

<FlatList

data={this.state.dataSource}

ItemSeparatorComponent={this.ListViewItemSeparator}

renderItem={({ item }) => (

<Text style={styles.textStyle}>{item.title}</Text>

)}

enableEmptySections={true}

style={{ marginTop: 10 }}

keyExtractor={(item, index) => index.toString()}

/>

</View>

);

}

}

const styles = StyleSheet.create({

viewStyle: {

justifyContent: 'center',

flex: 1,

marginTop: 40,

padding: 16,

},

textStyle: {

padding: 10,

},

textInputStyle: {

height: 40,

borderWidth: 1,

paddingLeft: 10,

borderColor: '#009688',

backgroundColor: '#FFFFFF',

},

});