**Explore the SectionList Component**

In an earlier video, you learned how to use the **SectionList** component to render large lists with section headers efficiently. In this reading, let’s explore the code from that video to build the **SectionList** component to display an extensive list of menu items for the Little Lemon app.

The code below contains the list of menu items you want to display on the Little Lemon menu. It is declared as an array of objects. A **title** categorizes each menu item. You have four categories: appetizers, main dishes, sides, and desserts.

Within each category, there is a **data** array that contains the name of each menu item for that category.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

const menuItemsToDisplay = [

  {

    title: 'Appetizers',

    data: [

      'Hummus',

      'Moutabal',

      'Falafel',

      'Marinated Olives',

      'Kofta',

      'Eggplant Salad',

    ],

  },

  {

    title: 'Main Dishes',

    data: ['Lentil Burger', 'Smoked Salmon', 'Kofta Burger', 'Turkish Kebab'],

  },

  {

    title: 'Sides',

    data: [

      'Fries',

      'Buttered Rice',

      'Bread Sticks',

      'Pita Pocket',

      'Lentil Soup',

      'Greek Salad',

      'Rice Pilaf',

    ],

  },

  {

    title: 'Desserts',

    data: ['Baklava', 'Tartufo', 'Tiramisu', 'Panna Cotta'],

  },

];

To use the **SectionList** component, you will need to import it directly from the React Native package since it is a core component.

1

import { View, Text, StyleSheet, SectionList } from'react-native';

The **SectionList** component has two required props you will need to pass as a bare minimum. It is very similar to the **FlatList** component you learned about in an earlier section of the course.

The first prop is the **sections** that accept a plain array. This array contains the list of items to display. The second required prop is the **renderItem***.* The **renderItem** takes an item from the **sections** and renders it into the list.

Now within the **MenuItems** component, you should render the **SectionList** component as follows:

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

1

2

  const renderSectionHeader = ({ section: { title } }) => (

    <Text style={menuStyles.sectionHeader}>{title} </Text>

  );

  return (

    <View style={menuStyles.container}>

      <SectionList

        keyExtractor={(item, index) => item + index}

        sections={menuItemsToDisplay}

        renderItem={renderItem}

        renderSectionHeader={renderSectionHeader}

       </SectionList>

    </View>

  );

};

const MenuItems = () => {

  const renderItem = ({ item }) => <Item name={item} />;

Notice here that you have provided the **sections** prop with the name of the array you already have defined, which is **menuItemsToDisplay***.* The **renderItem** method calls a call-back method which renders another component called **Item***.* You can also see an additional prop, the **keyExtractor** being passed to the **SectionList** component. It tells the list to use each **id** as React keys.

Notice there is a **renderSectionHeader** call-back function passed to the prop as well. This is used to render each section’s header. These examples would be Appetizers, Main Dishes and so on. It renders **title** from the **menuItemsToDisplay** array.

Let’s define the **Item** component next:

5

);

In the above code, the **Item** component takes **name** as the prop and it renders **name** within a parent view. The **Item** component is rendered for every item in the array until it reaches the end.

You can enhance your **SectionList** component with separators and footers if you want. They are passed to the component via the **ListFooterComponent** prop and the **ItemSeparatorComponent** prop as shown below:

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

  const renderSectionHeader = ({ section: { title } }) => (

    <Text style={menuStyles.sectionHeader}>{title} </Text>

  );

  return (

    <View style={menuStyles.container}>

      <SectionList

        keyExtractor={(item, index) => item + index}

        sections={menuItemsToDisplay}

        renderItem={renderItem}

        renderSectionHeader={renderSectionHeader}

        ListFooterComponent={Footer}

        ItemSeparatorComponent={Separator}></SectionList>

    </View>

  );

};

Below you'll find the component code for the separator and footer:

1

2

3

4

5

6

7

const Separator = () => <View style={menuStyles.separator} />;

const Footer = () => (

  <Text style={menuStyles.footerText}>

    All Rights Reserved by Little Lemon 2022

  </Text>

);

And that’s it!

The **SectionList** renders items lazily, which means it only renders items you visually see on the screen, and once you start scrolling up or down, those items off the screen are removed, and the new items are rendered. This form of lazy rendering is very performant and effective in rendering large lists with sectional support in mobile apps.

Below are the styles used in this component for your reference:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

const menuStyles = StyleSheet.create({

  container: {

    flex: 0.95,

  },

  innerContainer: {

    paddingHorizontal: 40,

    paddingVertical: 20,

    backgroundColor: '#333333',

  },

  sectionHeader: {

    backgroundColor: '#fbdabb',

    color: '#333333',

    fontSize: 34,

    flexWrap: 'wrap',

    textAlign: 'center',

  },

  itemText: {

    color: '#F4CE14',

    fontSize: 32,

  },

  separator: {

    borderBottomWidth: 1,

    borderColor: '#EDEFEE',

  },

  footerText: {

    color: '#EDEFEE',

    fontSize: 20,

    flexWrap: 'wrap',

    textAlign: 'center',

  },

});

Putting it all together below is the entire **MenuItems** component that displays the lists by sections.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

import React from 'react';

import { View, Text, StyleSheet, SectionList } from 'react-native';

const menuItemsToDisplay = [

  {

    title: 'Appetizers',

    data: [

      'Hummus',

      'Moutabal',

      'Falafel',

      'Marinated Olives',

      'Kofta',

      'Eggplant Salad',

    ],

  },

  {

    title: 'Main Dishes',

    data: ['Lentil Burger', 'Smoked Salmon', 'Kofta Burger', 'Turkish Kebab'],

  },

  {

    title: 'Sides',

    data: [

      'Fries',

      'Buttered Rice',

      'Bread Sticks',

      'Pita Pocket',

      'Lentil Soup',

      'Greek Salad',

      'Rice Pilaf',

    ],

  },

  {

    title: 'Desserts',

    data: ['Baklava', 'Tartufo', 'Tiramisu', 'Panna Cotta'],

  },

];

const Item = ({ name }) => (

  <View style={menuStyles.innerContainer}>

In this section, you learned how to use the **SectionList** component to render large lists with section headers in React Native efficiently.

Mark as completed

Like

Dislike

Report an issue