

1. Atividades Práticas - Usando ScrollView e List Views

Nesta atividade prática, vamos trabalhar com os conceitos relacionados ao ScrollView do React Native.

1.1 Introdução

Vimos que o ScrollView é um contêiner de rolagem genérico que pode hospedar vários componentes e Views. Todavia os itens roláveis não precisam ser homogêneos e você pode rolar verticalmente e horizontalmente. Também vimos anteriormente que um ScrollView renderiza todos os seus componentes filho de uma vez só. Porém, isso pode ter uma desvantagem de desempenho. Assim, o React Native fornece um conjunto de componentes para apresentar listas de dados. Geralmente, você deverá usar o FlatList ou o SectionList.

O componente FlatList exibe uma lista de rolagem de dados variáveis, mas estruturados de forma semelhante. O FlatList funciona bem para longas listas de dados, onde o número de itens pode mudar com o tempo. Entretanto, se você desejar renderizar um conjunto de dados dividido em seções lógicas, talvez com cabeçalhos de seção, então uma SectionList é a melhor opção.

1.2 Criando nosso Projeto

Vamos criar um projeto chamado my-style, digitando este comando no terminal:

```
expo init my-lists
```

Em seguida, algumas opções serão apresentadas. Selecione a opção “blank”. O Expo será executado no diretório que você especificar. Este processo leva alguns segundos e, logo após terminar, digite o seguinte comando para entrar no diretório recém criado do nosso projeto:

```
cd my-lists
```

Você agora está dentro da raiz do seu projeto. Até aqui, você criou um projeto e adicionou todas as dependências. Você pode, agora, abrir a aplicação no editor Visual Studio Code. Para isso, dentro do diretório `my-lists` digite o seguinte comando:

```
code .
```

O Visual Studio Code deverá abrir com a pasta raiz do seu projeto sendo acessada. Em seguida, você inicializará um servidor de desenvolvimento local. Assim, execute o seguinte comando:

```
expo start
```

Ao executar esse script, o Expo CLI inicia o Metro Bundler, que é um servidor HTTP que compila o código JavaScript de nosso aplicativo (usando o Babel) e o serve ao aplicativo Expo.

1.3 Usando o ScrollView

Primeiro de tudo, crie um um arquivo chamado ScrollViewExample.js com o seguinte código:

```
import React from "react";
import { Text, View, StyleSheet, ScrollView } from
"react-native";

export default function ScrollViewExample() {
  const state = {
    names: [
      { name: "Ben", id: 1 },
      { name: "Susan", id: 2 },
      { name: "Robert", id: 3 },
      { name: "Mary", id: 4 },
      { name: "Daniel", id: 5 },
      { name: "Laura", id: 6 },
      { name: "John", id: 7 },
      { name: "Debra", id: 8 },
      { name: "Aron", id: 9 },
      { name: "Ann", id: 10 },
    ],
  };
}
```

```

    { name: "Steve", id: 11 },
    { name: "Olivia", id: 12 },
  ],
};
return (
  <View>
    <ScrollView>
      {state.names.map((item, index) => (
        <View key={item.id} style={styles.item}>
          <Text>{item.name}</Text>
        </View>
      ))}
    </ScrollView>
  </View>
);
}
const styles = StyleSheet.create ({
  item: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    padding: 30,
    margin: 2,
    borderColor: '#2a4944',
    borderWidth: 1,
    backgroundColor: '#d2f7f1'
  }
});

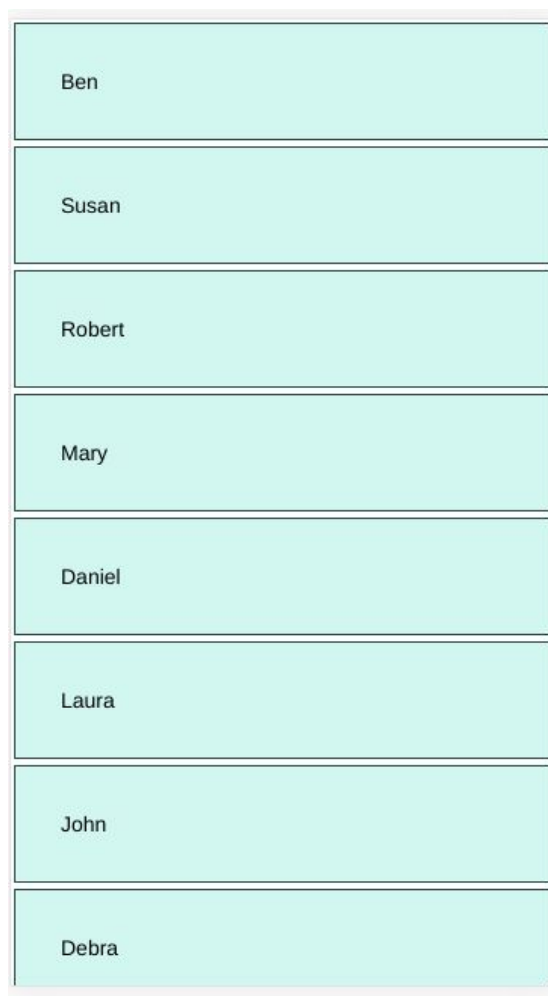
```

Agora, altere o arquivo App.js para chamar este componente.

```
import React from 'react';
import ScrollViewExample from './ScrollViewExample';

export default function App() {
  return (
    <ScrollViewExample />
  );
}
```

Veja o resultado na tela:



1.4 Usando o FlatList

Agora, crie um um arquivo chamado FlatListExample.js com o seguinte código:

```
import React, { useState } from "react";
import { FlatList, SafeAreaView, StatusBar, StyleSheet, Text,
TouchableOpacity } from "react-native";

const DATA = [
  {
    id: "bd7acbea-c1b1-46c2-aed5-3ad53abb28ba",
    title: "First Item",
  },
  {
    id: "3ac68afc-c605-48d3-a4f8-fbd91aa97f63",
    title: "Second Item",
  },
  {
    id: "58694a0f-3da1-471f-bd96-145571e29d72",
    title: "Third Item",
  },
];

const Item = ({ item, onPress, style }) => (
  <TouchableOpacity onPress={onPress} style={style}>
    <Text style={style.title}>{item.title}</Text>
  </TouchableOpacity>
);

const FlatListExample = () => {
  const [selectedId, setSelectedId] = useState(null);

  const renderItem = ({ item }) => {
```

```

    const backgroundColor = item.id === selectedId ? "#6e3b6e"
: "#f9c2ff";

    return (
      <Item
        item={item}
        onPress={() => setSelectedId(item.id)}
        style={{ backgroundColor }}
      />
    );
  };

  return (
    <SafeAreaView style={styles.container}>
      <FlatList
        data={DATA}
        renderItem={renderItem}
        keyExtractor={(item) => item.id}
        extraData={selectedId}
      />
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    marginTop: StatusBar.currentHeight || 0,
  },
  item: {

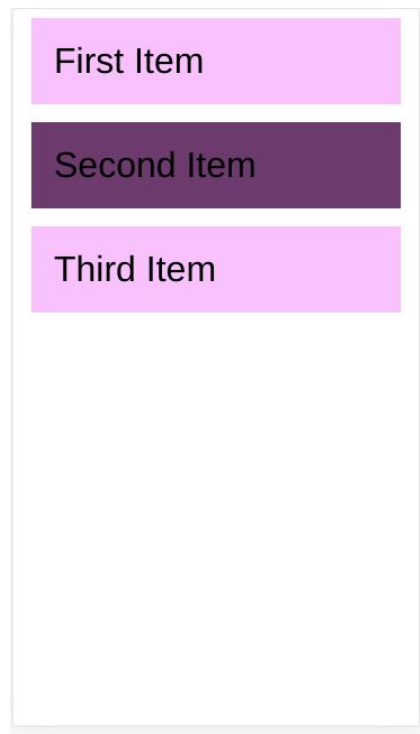
```

```
padding: 20,  
marginVertical: 8,  
marginHorizontal: 16,  
},  
title: {  
  fontSize: 32,  
},  
});  
  
export default FlatListExample;
```

Agora, altere o arquivo App.js para chamar este componente.

```
import React from 'react';  
import FlatListExample from './FlatListExample';  
  
export default function App() {  
  return (  
    <FlatListExample />  
  );  
}
```

Veja o resultado na tela ao clicar no segundo item da listagem:



1.5 Usando o SectionList

Agora, crie um um arquivo chamado SectionListExample.js com o seguinte código:

```
import React from "react";
import {
  StyleSheet,
  Text,
  View,
  SafeAreaView,
  SectionList
} from "react-native";
import Constants from "expo-constants";

const DATA = [
  {
    title: "Main dishes",
```



```

    data: ["Pizza", "Burger", "Risotto"]
  },
  {
    title: "Sides",
    data: ["French Fries", "Onion Rings", "Fried Shrimps"]
  },
  {
    title: "Drinks",
    data: ["Water", "Coke", "Beer"]
  },
  {
    title: "Desserts",
    data: ["Cheese Cake", "Ice Cream"]
  }
];

```

```

const Item = ({ title }) => (
  <View style={styles.item}>
    <Text style={styles.title}>{title}</Text>
  </View>
);

```

```

const SectionListExample = () => (
  <SafeAreaView style={styles.container}>
    <SectionList
      sections={DATA}
      keyExtractor={(item, index) => item + index}
      renderItem={({ item }) => <Item title={item} />}
      renderSectionHeader={({ section: { title } }) => (
        <Text style={styles.header}>{title}</Text>

```

```

    )}

    />
</SafeAreaView>
);

const styles = StyleSheet.create({
  container: {
    flex: 1,
    marginTop: Constants.statusBarHeight,
    marginHorizontal: 16
  },
  item: {
    backgroundColor: "#f9c2ff",
    padding: 20,
    marginVertical: 8
  },
  header: {
    fontSize: 32,
    backgroundColor: "#fff"
  },
  title: {
    fontSize: 24
  }
});

export default SectionListExample;

```

Agora, altere o arquivo App.js para chamar este componente.

```
import React from 'react';
import SectionListExample from './SectionListExample';

export default function App() {
  return (
    <SectionListExample />
  );
}
```

Veja o resultado na tela:

