

Lesson:

Refresh Control



The **RefreshControl** component in React Native is a built-in component that provides pull-to-refresh functionality for scrollable views, such as **ScrollView** and **FlatList**. It allows users to manually trigger a refresh of the content within the scrollable view.

Using the RefreshControl Component

To use the **RefreshControl** component, you can import it from the **react-native** library and include it as a prop in your scrollable view component. Here's an example of how to use the **RefreshControl** component with a **ScrollView**:

```
1 import React, { useState } from 'react';
2 import { View, ScrollView, RefreshControl, StyleSheet, Text } from 'react-native';
3
4 const App = () => {
5   const [refreshing, setRefreshing] = useState(false);
6
7   const onRefresh = () => {
8     setRefreshing(true);
9
10    // Perform data fetching or other asynchronous tasks
11    setTimeout(() => {
12      setRefreshing(false);
13    }, 2000);
14  };
15
16  return (
17    <ScrollView
18      contentContainerStyle={styles.container}
19      refreshControl={
20        <RefreshControl
21          refreshing={refreshing}
22          onRefresh={onRefresh}
23          tintColors="#FF0000"
24          title="Pull to Refresh"
25          titleColor="#FF0000"
26          progressBackgroundColor="#FFFFFF"
27        />
28      }
29    >
30      <Text>Scrollable content here</Text>
31    </ScrollView>
32  );
33 };
34
35 const styles = StyleSheet.create({
36   container: {
37     flex: 1,
38     justifyContent: 'center',
39     alignItems: 'center',
40   },
41 });
42
43 export default App;
```

In this example, we have wrapped our **ScrollView** component with the **RefreshControl** component. We have defined a state variable refreshing using the **useState** hook to keep track of the refreshing state. The **onRefresh** function is called when the user triggers the refresh.

The **RefreshControl** component is passed as the **refreshControl** prop to the **ScrollView**. We have set various props such as **refreshing**, **onRefresh**, **tintColor**, **title**, **titleColor**, and **progressBackgroundColor** to customize the appearance and behavior of the refresh control.

Scrollable content here

Props of RefreshControl

The **RefreshControl** component accepts several props to control its behavior and appearance. Here are some commonly used props:

- **refreshing** (boolean): Specifies whether the refresh control is in the refreshing state or not. Typically, you would update this prop based on the state of your data fetching or other asynchronous tasks.
- **onRefresh** (function): Callback function to be called when the user triggers the refresh. You can perform data fetching or other asynchronous tasks inside this function.
- **tintColor** (string): Sets the color of the refresh indicator when it is active. You can provide a color name or a hexadecimal code.
- **title** (string): Displays a text title next to the refresh indicator. Use this prop to provide a custom text message to prompt the user to pull to refresh.
- **titleColor** (string): Sets the color of the title text. You can provide a color name or a hexadecimal code.
- **progressBackgroundColor** (string): Sets the background color behind the refresh indicator. You can provide a color name or a hexadecimal code.

These are just a few of the available props for the **RefreshControl** component. You can explore the React Native documentation for more advanced options and customization.

Use Cases

The RefreshControl component is useful in scenarios where you have scrollable content that can be refreshed manually by the user. Some common use cases include:

- Pull-to-refresh functionality: Enable users to pull down the content to trigger a refresh, commonly found in news or social media applications.
- Refreshing data from an API: Fetch and update data from an API when the user triggers a refresh.
- Updating local data: Refreshing and updating local data or states based on user interaction.

Default Prop Values

The default values for the props of the **RefreshControl** component are as follows:

- **refreshing:** false
- **onRefresh:** undefined
- **tintColor:** undefined
- **title:** undefined
- **titleColor:** undefined
- **progressBackgroundColor:** undefined

These default values can be overridden by passing the corresponding props to the RefreshControl component.

By utilizing the **RefreshControl** component in your React Native application, you can enhance the user experience by allowing them to manually trigger a refresh of the scrollable content, providing them with the most up-to-date information.