

Lesson:

Activity Indicator



The `ActivityIndicator` component in React Native provides a visual loading indicator that can be displayed while waiting for content to load or perform a time-consuming task. It is a built-in component that allows you to show an animated loading spinner to indicate that something is happening in the background.

Using the ActivityIndicator Component

To use the `ActivityIndicator` component, you simply need to include it in your component's JSX and conditionally show/hide it based on the loading state. Here's an example:

```

1  import React, { useState, useEffect } from 'react';
2  import { View, ActivityIndicator, StyleSheet, Text } from 'react-native';
3
4  const App = () => {
5    const [isLoading, setIsLoading] = useState(true);
6
7    useEffect(() => {
8      // Simulating an API call or time-consuming task
9      setTimeout(() => {
10        setIsLoading(false);
11      }, 3000);
12    }, []);
13
14    return (
15      <View style={styles.container}>
16        {isLoading ? (
17          <ActivityIndicator size="large" color="#0000ff" />
18        ) : (
19          <Text>Data loaded successfully!</Text>
20        )}
21      </View>
22    );
23  };
24
25  const styles = StyleSheet.create({
26    container: {
27      flex: 1,
28      justifyContent: 'center',
29      alignItems: 'center',
30    },
31  });
32
33  export default App;

```

In this example, we have a simple app that displays an `ActivityIndicator` component while data is being loaded. We use the `useState` hook to manage the loading state and the `useEffect` hook to simulate a loading delay. When the loading state is true, the `ActivityIndicator` is displayed; otherwise, the "Data loaded successfully!" message is shown.



Props of ActivityIndicator

The `ActivityIndicator` component provides several props that allow you to customize its appearance and behavior. Here are the commonly used props with their default values:

- **animating** (boolean): Controls whether the activity indicator is animating. By default, it is set to `true`.
- **color** (string): Sets the color of the activity indicator. The default color is platform-dependent.
- **size** (string or number): Specifies the size of the activity indicator. It accepts values such as "small", "large", or a specific number representing the size in pixels.
- **hidesWhenStopped** (boolean): Determines whether the activity indicator is hidden when it is not animating. By default, it is set to `true`.

Use Cases

The `ActivityIndicator` component is useful in various scenarios where you want to provide visual feedback to users during loading or time-consuming tasks. Here are some common use cases for using

ActivityIndicator:

- **Data fetching:** Display the activity indicator while fetching data from an API or performing asynchronous operations.
- **Image loading:** Show the activity indicator while loading and rendering images.
- **Form submission:** Indicate that the form submission is in progress and prevent user interactions until the submission is complete.
- **Content loading:** Display the activity indicator when loading content, such as loading a web page or rendering complex UI components.

By using the **ActivityIndicator** component, you can provide a clear visual indication to users that something is happening in the background, enhancing the user experience and managing user expectations during loading or time-consuming processes.