

# Lesson:

# Status Bar



The **StatusBar** component in React Native is a built-in component that allows you to control the appearance of the status bar at the top of your app screen. The status bar displays important system information, such as the time, battery level, and network connectivity.

## Using the StatusBar Component

To use the **StatusBar** component, you can import it from the **react-native** library and include it in your app's render method. Here's an example of how to use the **StatusBar** component to customize the status bar:

```
1 import React from 'react';
2 import { View, StatusBar, StyleSheet } from 'react-native';
3
4 const App = () => {
5   return (
6     <View style={styles.container}>
7       <StatusBar
8         backgroundColor="#FF0000"
9         barStyle="dark-content"
10        hidden={false}
11      />
12      /* Rest of your app content */
13    </View>
14  );
15 };
16
17 const styles = StyleSheet.create({
18   container: {
19     flex: 1,
20     justifyContent: 'center',
21     alignItems: 'center',
22   },
23 });
24
25 export default App;
```

In this example, we have included the **StatusBar** component inside the main **View** component of our app. We have set the background color to red (**#FF0000**), the text color to light content, and made the status bar visible (**hidden={false}**). You can customize these properties based on your app's design and requirements.

12:15



# Props of StatusBar

The StatusBar component accepts several props to customize its appearance and behavior. Here are some commonly used props:

- **backgroundColor** (string): Specifies the background color of the status bar. You can use color names (e.g., "red"), hexadecimal codes (e.g., "#FF0000"), or other valid color representations.
- **barStyle** (string): Specifies the style of the status bar text. You can set it to "default" for dark text on a light background, "light-content" for light text on a dark background, or "dark-content" for dark text on a light background.
- **hidden** (boolean): Controls the visibility of the status bar. Set it to true to hide the status bar or false to make it visible.
- **translucent** (boolean): Specifies whether the status bar is translucent or not. When set to true, the status bar overlays the app content. Default value is false.
- **networkActivityIndicatorVisible** (boolean): Controls the visibility of the network activity indicator in the status bar. Set it to true to show the activity indicator or false to hide it.

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

## Use Cases

The **StatusBar** component is useful in various scenarios where you want to customize the appearance of the status bar or control its visibility. Here are some common use cases:

- Design consistency: Set the status bar color to match your app's overall color scheme, providing a consistent visual experience.
- Dark mode support: Adjust the status bar text color based on the app's theme, ensuring readability in both light and dark mode.
- Full-screen mode: Hide the status bar when displaying full-screen content, such as images, videos, or games, to provide a more immersive user experience.
- Network activity indication: Show the network activity indicator in the status bar to inform users when the app is making network requests or loading data.
- Custom status bar styles: Customize the status bar appearance for specific screens or components within your app to differentiate their visual presentation.

By using the StatusBar component, you can enhance the user experience by fine-tuning the appearance and behavior of the status bar to align with your app's design and functionality.