

Modal

The Modal component is a basic way to present content above an enclosing view.

Example

Modal

^ Expo

```
import React, {useState} from 'react';
import {Alert, Modal, StyleSheet, Text, Pressable, View} from 'react-native';

const App = () => {
  const [modalVisible, setModalVisible] = useState(false);
  return (
    <View style={styles.centeredView}>
      <Modal
        animationType="slide"
        transparent={true}
        visible={modalVisible}
        onRequestClose={() => {
          Alert.alert('Modal has been closed.');
          setModalVisible(!modalVisible);
        }}>
        <View style={styles.centeredView}>
          <View style={styles.modalView}>
            <Text style={styles.modalText}>Hello World!</Text>
            <Pressable
              style={[styles.button, styles.buttonClose]}
              onPress={() => setModalVisible(!modalVisible)}>
              <Text style={styles.textStyle}>Hide Modal</Text>
            </Pressable>
          </View>
        </View>
      </View>
    </View>
  );
};
```

Preview

My Device

iOS

Android

Reference

Props

View Props

Inherits View Props.

animated

Deprecated. Use the animationType prop instead.

animationType

The `animationType` prop controls how the modal animates.

Possible values:

- `slide` slides in from the bottom,
- `fade` fades into view,
- `none` appears without an animation.

TYPE	DEFAULT
<code>enum('none', 'slide', 'fade')</code>	<code>none</code>

hardwareAccelerated Android

The `hardwareAccelerated` prop controls whether to force hardware acceleration for the underlying window.

TYPE	DEFAULT
<code>bool</code>	<code>false</code>

onDismiss iOS

The `onDismiss` prop allows passing a function that will be called once the modal has been dismissed.

TYPE
function

`onOrientationChange` ◀ iOS

The `onOrientationChange` callback is called when the orientation changes while the modal is being displayed. The orientation provided is only 'portrait' or 'landscape'. This callback is also called on initial render, regardless of the current orientation.

TYPE
function

`onRequestClose`

The `onRequestClose` callback is called when the user taps the hardware back button on Android or the menu button on Apple TV. Because of this required prop, be aware that `BackHandler` events will not be emitted as long as the modal is open. On iOS, this callback is called when a Modal is being dismissed using a drag gesture when `presentationStyle` is `pageSheet` or `formSheet`

TYPE
function Required ◀ Android ◀ TV
function ◀ iOS

`onShow`

The `onShow` prop allows passing a function that will be called once the modal has been shown.

TYPE
function

presentationStyle ◀ iOS

The `presentationStyle` prop controls how the modal appears (generally on larger devices such as iPad or plus-sized iPhones). See <https://developer.apple.com/reference/uikit/uimodalpresentationstyle> for details.

Possible values:

- `fullscreen` covers the screen completely
- `pageSheet` covers portrait-width view centered (only on larger devices)
- `formSheet` covers narrow-width view centered (only on larger devices)
- `overFullScreen` covers the screen completely, but allows transparency

TYPE	DEFAULT
<code>enum('fullscreen', 'pageSheet', 'formSheet', 'overFullScreen')</code>	<code>fullscreen</code> if <code>transparent={false}</code> <code>overFullScreen</code> if <code>transparent={true}</code>

statusBarTranslucent ▶ Android

The `statusBarTranslucent` prop determines whether your modal should go under the system statusbar.

TYPE	DEFAULT
bool	false

TYPE	DEFAULT

supportedOrientations ◀ iOS

The `supportedOrientations` prop allows the modal to be rotated to any of the specified orientations. On iOS, the modal is still restricted by what's specified in your app's Info.plist's `UISupportedInterfaceOrientations` field.

When using `presentationStyle` of `pageSheet` or `formSheet`, this property will be ignored by iOS.

TYPE	DEFAULT
array of enums('portrait', 'portrait-upside-down', 'landscape', 'landscape-left', 'landscape-right')	['portrait']

transparent

The `transparent` prop determines whether your modal will fill the entire view. Setting this to `true` will render the modal over a transparent background.


TYPE	DEFAULT
bool	false

visible

The `visible` prop determines whether your modal is visible.

TYPE	DEFAULT
bool	true

Is this page useful?  

 Edit this page

*Last updated on **Jun 21, 2023***