# **TouchableNativeFeedback**

If you're looking for a more extensive and future-proof way to handle touch-based input, check out the Pressable API.

A wrapper for making views respond properly to touches (Android only). On Android this component uses native state drawable to display touch feedback.

At the moment it only supports having a single View instance as a child node, as it's implemented by replacing that View with another instance of RCTView node with some additional properties set.

Background drawable of native feedback touchable can be customized with background property.

## **Example**

#### **∧** Expo TouchableNativeFeedback Android Component Example import React, {useState} from 'react'; Download Expo Go and scan import { Text, the QR code to get started. View, StyleSheet, TouchableNativeFeedback, StatusBar, } from 'react-native'; const App = () => { const [rippleColor, setRippleColor] = useState(randomHexColor()); const [rippleOverflow, setRippleOverflow] = useState(false); return ( <View style={styles.container}> <TouchableNativeFeedback</pre> onPress={() => { setRippleColor(randomHexColor()); setRippleOverflow(!rippleOverflow); **Connected devices** background={TouchableNativeFeedback.Ripple( rippleColor, rippleOverflow, <u>Log in</u> or set a <u>Device ID</u> to open this Snack )}> from Expo Go on your device or simulator. <View style={styles.touchable}> Preview My Device Android

# Reference

### **Props**

### TouchableWithoutFeedback Props

Inherits TouchableWithoutFeedback Props.

### background

Determines the type of background drawable that's going to be used to display feedback. It takes an object with type property and extra data depending on the type. It's recommended to use one of the static methods to generate that dictionary.

TYPE	
backgroundPropType	

#### useForeground

Set to true to add the ripple effect to the foreground of the view, instead of the background. This is useful if one of your child views has a background of its own, or you're e.g. displaying images, and you don't want the ripple to be covered by them.

Check TouchableNativeFeedback.canUseNativeForeground() first, as this is only available on Android 6.0 and above. If you try to use this on older versions you will get a warning and fallback to background.

TYPE	
bool	

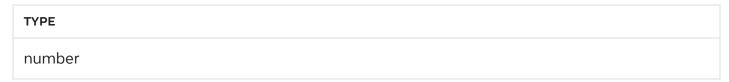
## 

TV preferred focus (see documentation for the View component).

TYPE		
bool		

# nextFocusDown Android

TV next focus down (see documentation for the View component).



## 

TV next focus forward (see documentation for the View component).

TYPE
number

## nextFocusLeft | Android

TV next focus left (see documentation for the View component).

TYPE number

# 

TV next focus right (see documentation for the View component).

TYPE number

#### 

TV next focus up (see documentation for the View component).

TYPE number

#### **Methods**

#### SelectableBackground()

```
static SelectableBackground(
    rippleRadius: number | null,
): ThemeAttributeBackgroundPropType;
```

Creates an object that represents android theme's default background for selectable elements (?android:attr/selectableItemBackground). rippleRadius parameter controls the radius of the ripple effect.

#### SelectableBackgroundBorderless()

```
static SelectableBackgroundBorderless(
    rippleRadius: number | null,
): ThemeAttributeBackgroundPropType;
```

Creates an object that represent android theme's default background for borderless selectable elements (?android:attr/selectableItemBackgroundBorderless). Available on android API level 21+. rippleRadius parameter controls the radius of the ripple effect.

#### Ripple()

```
static Ripple(
  color: ColorValue,
  borderless: boolean,
```

```
rippleRadius?: number | null,
): RippleBackgroundPropType;
```

Creates an object that represents ripple drawable with specified color (as a string). If property borderless evaluates to true the ripple will render outside of the view bounds (see native actionbar buttons as an example of that behavior). This background type is available on Android API level 21+.

#### **Parameters:**

NAME	TYPE	REQUIRED	DESCRIPTION
color	string	Yes	The ripple color
borderless	boolean	Yes	If the ripple can render outside its bounds
rippleRadius	?number	No	controls the radius of the ripple effect

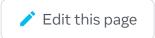
#### canUseNativeForeground()

static canUseNativeForeground(): boolean;

# Is this page useful?







Last updated on **Sep 3, 2023**