

Version: 3.x

dispatchCommand

! INFO

This page was ported from an old version of the documentation.

As we're rewriting the documentation some of the pages might be a little outdated.

Allows to dispatch command on a native component synchronously from the UI thread.

Arguments

`animatedRef`

The product of `useAnimatedRef` which is Reanimated's extension of a standard React ref (delivers the view tag on the UI thread).

`commandName` **[String]**

The name of the command to execute, e.g. `focus` or `scrollToEnd`.

`args` **[Array<any>]**

The array of command arguments. Defaults to an empty array.

Returns

`void`

Example

```
import Animated, {
  dispatchCommand,
  useAnimatedRef,
} from 'react-native-reanimated';
import { Button, StyleSheet, TextInput, View } from 'react-native';
import { Gesture, GestureDetector } from 'react-native-gesture-handler';

import React from 'react';

const AnimatedTextInput = Animated.createAnimatedComponent(TextInput);

export default function DispatchCommandExample() {
  const aref = useAnimatedRef<TextInput>();

  const gesture = Gesture.Tap().onStart(() => {
    dispatchCommand(aref, 'focus');
  });

  return (
    <View style={styles.container}>
      <AnimatedTextInput ref={aref} style={styles.input} />
      <GestureDetector gesture={gesture}>
        <Button title="Focus" />
      </GestureDetector>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
  },
  input: {
    borderWidth: 1,
    width: 200,
    padding: 5,
  },
});
```

 [Edit this page](#)