Version: 3.x

useAnimatedRef

`useAnimatedRef` lets you get a reference of a view. Used alongside `measure`, `scrollTo`, and `useScrollViewOffset` functions.

An object defined using `useAnimatedRef` has to be passed to the `ref` property of a component.

Reference

```
import { useAnimatedRef } from 'react-native-reanimated';

function App() {
  const animatedRef = useAnimatedRef();

  return <Animated.View ref={animatedRef} />;
}
```

Type definitions

Arguments

`useAnimatedRef` doesn't take any arguments.

Returns

`useAnimatedRef` returns an object with a `current` property which contains an instance of a component.

Example

Expand the full code

```
export default function App() {
  const animatedRef = useAnimatedRef();

  return (
    <View style={styles.container}>
      <Animated.View
        ref={animatedRef}
        style={styles.box}
        onLayout={() => {
          // Returns a reference to the component
          const component = animatedRef.current;
        }}
      />
    </View>
  );
}
```

Remarks

- You can use `useAnimatedRef` to reference not only the Animated versions of components, but any React Native component.
- The value stored in the current property becomes available after the component is mounted.

```
function App() {
  const animatedRef = useAnimatedRef();

  console.log(animatedRef.current); // 🚩 Returns null

  useEffect(() => {
    console.log(animatedRef.current); // ✅ Returns the component
  }, []);

  return <View ref={animatedRef} />;
}
```

Alternatively, you can get the value stored in the `current` in **event handlers** or in a `onLayout` property.

- The value stored in the `current` property isn't available on the UI thread.

Platform compatibility

Android	iOS	Web
✓	✓	✓

`

Edit this page