Version: 3.x

useFrameCallback

`useFrameCallback` lets you run a function on every frame update.

Preview Code

Reference

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

function App() {
  const frameCallback = useFrameCallback((frameInfo) => {
    // Increment a value on every frame update
    sv.value += 1;
  });

  return (
    <Button
      title="Start/Stop"
      onPress={() => frameCallback.setActive(!frameCallback.isActive)}
    />
  );
}
```

Type definitions

Type definitions

Arguments

`callback`

A function executed on every frame update. This function receives a `frameInfo` object containing the following fields:

- `timestamp` a number indicating the system time (in milliseconds) when the last frame was rendered.
- `timeSincePreviousFrame` a number indicating the time (in milliseconds) since last frame. This value will be null on the first frame after activation. Starting from the second frame, it should be ~16 ms on 60 Hz, and ~8 ms on 120 Hz displays (provided there are no frame dropped).
- `timeSinceFirstFrame` a number indicating the time (in milliseconds) since the callback was activated.

`autostart` Optional

Whether the callback should start automatically. Defaults to `true`.

Returns

`useFrameCallback` returns an object containing these fields:

- `setActive` a function that lets you start the frame callback or stop it from running
- `isActive` a boolean indicating whether a callback is running
- `callbackId` a number indicating a unique identifier of the frame callback

Example

Preview Code

Tap to jump

Remarks

- A function passed to the `callback` argument is automatically workletized and ran on the UI thread.

Platform compatibility

Android	iOS	Web
✅	✅	✅

Edit this page