

# Systrace

Systrace is a standard Android marker-based profiling tool (and is installed when you install the Android platform-tools package). Profiled code blocks are surrounded by start/end markers which are then visualized in a colorful chart format. Both the Android SDK and React Native framework provide standard markers that you can visualize.

## Example

Systrace allows you to mark JavaScript (JS) events with a tag and an integer value. Capture the non-Timed JS events in EasyProfiler.

Systrace Example

^ Expo

```
import React from 'react';
import {
  Button,
  Text,
  View,
  SafeAreaView,
  StyleSheet,
  Systrace,
} from 'react-native';

const App = () => {
  const enableProfiling = () => {
    Systrace.setEnabled(true); // Call setEnabled to turn
on the profiling.
    Systrace.beginEvent('event_label');
    Systrace.counterEvent('event_label', 10);
  };

  const stopProfiling = () => {
    Systrace.endEvent();
  };

  return (
    <SafeAreaView style={styles.container}>
      <Text style={[styles.header, styles.paragraph]}>
        React Native Systrace API
      </Text>
    </SafeAreaView>
  );
};
```

Connecting...

Preview ☐

My Device

iOS

Android

Web

# Reference

## Methods

### isEnabled()

```
static isEnabled(): boolean;
```

### beginEvent()

```
static beginEvent(eventName: string | (() => string), args?: EventArgs);
```

beginEvent/endEvent for starting and then ending a profile within the same call stack frame.

### endEvent()

```
static endEvent(args?: EventArgs);
```

### beginAsyncEvent()

```
static beginAsyncEvent(  
  eventName: string | (() => string),  
  args?: EventArgs,  
): number;
```

beginAsyncEvent/endAsyncEvent for starting and then ending a profile where the end can either occur on another thread or out of the current stack frame, eg await the

returned cookie variable should be used as input into the `endAsyncEvent` call to end the profile.

## `endAsyncEvent()`


```
static endAsyncEvent(  
  eventName: EventName,  
  cookie: number,  
  args?: EventArgs,  
);
```

## `counterEvent()`

```
static counterEvent(eventName: string | (() => string), value: number);
```

Register the value to the `profileName` on the systrace timeline.

Is this page useful?  

 Edit this page

Last updated on **Jun 21, 2023**