

How do I track screen views in a React Native app that uses Expo Router?



Sensor: Conviva React Native App Analytics ([@convivainc/conviva-react-native-appanalytics](#))

Short answer: Expo Router does not work with Conviva's existing screen-view auto-tracking helper. Track screen views manually by calling `trackScreenViewEvent()` from your root layout.

WHY AUTO-TRACKING DOES NOT APPLY

Conviva's React Native sensor auto-tracks screen views using the `withReactNavigationAutotrack` helper, which works by wrapping React Navigation's `NavigationContainer`.

Expo Router manages the `NavigationContainer` internally and does not expose it to the application code (per Expo's own [migration guide](#)). Because the helper has no `NavigationContainer` to wrap, it cannot be applied in an Expo Router app, and screen views are not auto-collected.

RECOMMENDED SOLUTION — MANUAL TRACKING

Following Expo's [screen tracking guide](#), call `trackScreenViewEvent()` from the app's root layout, passing the current screen `name`.

Sample code

```
// app/_layout.tsx
import { useEffect } from 'react';
import { usePathname, useSegments, useGlobalSearchParams, Slot } from 'expo-router';
import { tracker } from '../convivaTracker'; // your createTracker() instance

export default function RootLayout() {
  const pathname = usePathname();
  const segments = useSegments();
  const params = useGlobalSearchParams();

  useEffect(() => {
    // Option A – resolved URL (e.g. "/profile/alice")
    const screenName = pathname;

    // Option B – route pattern (e.g. "profile/[user]")
    // const screenName = segments.join('/') || 'index';

    tracker.trackScreenViewEvent({ name: screenName });
  }, [pathname, params]);

  return <Slot />;
}
```

Implementation Details & Reference

CHOOSING THE SCREEN NAME — TWO OPTIONS

OPTION	WHAT IT RETURNS	WHEN TO USE
<code>usePathname()</code>	Resolved URL (e.g. <code>/profile/</code> <code>alice</code>)	Individual paths are meaningful in dashboards; cardinality stays bounded by actual routes users visit.
<code>useSegments()</code>	Route pattern (e.g. <code>profile/</code> <code>[user]</code>)	App has many dynamic routes (user IDs, product IDs); want all visits to roll up under one screen name. Closer to how React Navigation's <code>screen_view.name</code> looks today.

Pick whichever better matches how you want screens to appear in Pulse reports.

NOTE ON THE PARAMS DEPENDENCY

Include `params` in the `useEffect` dependency array only if you want a new `screen_view` event to fire when query-string parameters change (for example `?tab=posts` → `?tab=likes`). Otherwise `[pathname]` alone is sufficient.

RELATED LINKS

- Conviva React Native sensor: github.com/Conviva/conviva-react-native-appanalytics
- Expo Router screen tracking guide: docs.expo.dev/router/reference/screen-tracking
- Expo Router → React Navigation migration: docs.expo.dev/router/migrate/from-react-navigation