Guides Screen tracking for analytics

Version: 6.x

Screen tracking for analytics

To track the currently active screen, we need to:

1. Add a callback to get notified of state changes
2. Get the root navigator state and find the active route name

To get notified of state changes, we can use the `onStateChange` prop on `NavigationContainer`. To get the root navigator state, we can use the `getRootState` method on the container's ref. Please note that `onStateChange` is not called on initial render so you have to set your initial screen separately.

Example

This example shows how the approach can be adapted to any mobile analytics SDK.

```
import {
  NavigationContainer,
  useNavigationContainerRef,
} from '@react-navigation/native';

export default () => {
  const navigationRef = useNavigationContainerRef();
  const routeNameRef = useRef();

  return (
    <NavigationContainer
      ref={navigationRef}
      onReady={() => {
        routeNameRef.current = navigationRef.getCurrentRoute().name;
      }}
      onStateChange={async () => {
        const previousRouteName = routeNameRef.current;
        const currentRouteName = navigationRef.getCurrentRoute().name;
        const trackScreenView = () => {
          // Your implementation of analytics goes here!
        };

        if (previousRouteName !== currentRouteName) {
          // Save the current route name for later comparison
          routeNameRef.current = currentRouteName;

          // Replace the line below to add the tracker from a mobile analytics
SDK
          await trackScreenView(currentRouteName);
        }
      }}
    >
      {/* ... */}
    </NavigationContainer>
  );
};
```

Try this example on Snack

Edit this page