



Version: 6.x

useNavigationState

`useNavigationState` is a hook which gives access to the `navigation state` of the navigator which contains the screen. It's useful in rare cases where you want to render something based on the navigation state.

Note: Consider the navigator's state object to be internal and subject to change in a minor release. Avoid using properties from the `navigation state` object except `index` and `routes`, unless you really need it. If there is some functionality you cannot achieve without relying on the structure of the state object, please open an issue.

It takes a selector function as an argument. The selector will receive the full `navigation state` and can return a specific value from the state:

```
const index = useNavigationState(state => state.index);
```

The selector function helps to reduce unnecessary re-renders, so your screen will re-render only when that's something you care about. If you actually need the whole state object, you can do this explicitly:

```
const state = useNavigationState(state => state);
```

Note: This hook is useful for advanced cases and it's easy to introduce performance issues if you're not careful. For most of the cases, you don't need the navigator's state.

How is `useNavigationState` different from `navigation.getState()`?

The `navigation.getState()` function also returns the current `navigation state`. The main difference is that the `useNavigationState` hook will trigger a re-render when values change, while `navigation.getState()` won't. For example, the following code will be incorrect:

```
function Profile() {  
  const routesLength = navigation.getState().routes.length; // Don't do this  
  
  return <Text>Number of routes: {routesLength}</Text>;  
}
```

In this example, even if you push a new screen, this text won't update. If you use the hook, it'll work as expected:

```
function Profile() {  
  const routesLength = useNavigationState(state => state.routes.length);  
  
  return <Text>Number of routes: {routesLength}</Text>;  
}
```


Try this example on Snack [↗](#)

So when do you use `navigation.getState()`? It's mostly useful within event listeners where you don't care about what's rendered. In most cases, using the hook should be preferred.

Using with class component

You can wrap your class component in a function component to use the hook:

```
class Profile extends React.Component {  
  render() {  
    // Get it from props  
    const { routesLength } = this.props;  
  }  
}  
  
// Wrap and export  
export default function(props) {  
  const routesLength = useNavigationState(state => state.routes.length);  
  
  return <Profile {...props} routesLength={routesLength} />;  
}
```

 [Edit this page](#)