



API reference

Hooks

useIsFocused

Version: 6.x

# useIsFocused

We might want to render different content based on the current focus state of the screen. The library exports a `useIsFocused` hook to make this easier:

```
import { useIsFocused } from '@react-navigation/native';

// ...

function Profile() {
  const isFocused = useIsFocused();

  return <Text>{isFocused ? 'focused' : 'unfocused'}</Text>;
}
```

Try this example on Snack [↗](#)

Note that using this hook triggers a re-render for the screen when it changes focus. This might cause lags during the animation if your screen is heavy. You might want to extract the expensive parts to separate components and use `React.memo` or `React.PureComponent` to minimize re-renders for them.


## 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 { isFocused } = this.props;
  }
}

// Wrap and export
export default function(props) {
```

```
const isFocused = useIsFocused();  
  
return <Profile {...props} isFocused={isFocused} />;  
}
```

 [Edit this page](#)