Guides Redux integration

Version: 6.x

Redux integration

It is extremely easy to use Redux in an app with React Navigation. It's basically no different than without React Navigation.

```
import { Provider } from 'react-redux';
import { NavigationContainer } from '@react-navigation/native';

// Render the app container component with the provider around it
export default function App() {
  return (
    <Provider store={store}>
      <NavigationContainer>
        {/* Screen configuration */}
      </NavigationContainer>
    </Provider>
  );
}
```

Notice that we wrap our components in a `Provider` like we'd normally do with `react-redux`. Ta da! Now feel free to use `connect` throughout your app.

Use a component that is `connect`ed in `options`

Create a component, `connect` it to the store, then use that component in the `title`.

```
function Counter({ value }) {
  return <Text>Count: {value}</Text>;
}

const CounterContainer = connect(state => ({ value: state.count }))(Counter);
```

Try this example on Snack

```
<Stack.Screen
  name="Test"
  component={TestScreen}
  options={{ title: () => <CounterContainer /> }}
/>
```

Pass the state you care about as a param to the screen

If the value isn't expected to change, you can just pass it from a connected component to the other screen as a param.

```
<Button
  title="Go to static counter screen"
  onPress={() =>
    props.navigation.navigate('StaticCounter', {
      count,
    })
  }
/>
```

```
function StaticCounter({ route }) {
  return (
    <View style={styles.container}>
      <Text style={styles.paragraph}>{route.params.count}</Text>
    </View>
  );
}
```

So our component will look like this:

```
<RootStack.Screen
  name="StaticCounter"
  component={StaticCounter}
  options={({ route }) => ({ title: route.params.count })}
/>
```

Try this example on Snack

Can I store the navigation state in Redux too?

This is not possible. We don't support it because it's too easy to shoot yourself in the foot and slow down / break your app.

However it's possible to use `redux-devtools-extension` to inspect the navigation state and actions, as well as perform time travel debugging by using the `devtools` package.

Edit this page