



Version: 6.x

Options for screens

Each screen can configure various aspects about how it gets presented in the navigator that renders it by specifying certain options, for example, the header title in stack navigator, tab bar icon in bottom tab navigator etc. Different navigators support different set of options.

In the [configuring the header bar](#) section of the fundamentals documentation we explain the basics of how this works. Also see the [screen options resolution guide](#) to get an idea of how they work when there are multiple navigators.

There are 3 ways of specifying options for screens:

options prop on Screen

You can pass a prop named `options` to the `Screen` component to configure a screen, where you can specify an object with different options for that screen:

```
<Stack.Navigator>
  <Stack.Screen
    name="Home"
    component={HomeScreen}
    options={{ title: 'Awesome app' }}
  />
  <Stack.Screen
    name="Profile"
    component={ProfileScreen}
    options={{ title: 'My profile' }}
  />
</Stack.Navigator>
```

Try this example on Snack [↗](#)

You can also pass a function to `options`. The function will receive the `navigation` prop and the `route` prop for that screen. This can be useful if you want to perform navigation in your options:

```
<Stack.Screen
  name="Home"
  component={HomeScreen}
  options={({ navigation }) => ({
    title: 'Awesome app',
    headerLeft: () => (
      <DrawerButton onPress={() => navigation.toggleDrawer()} />
    ),
  })}
/>
```

screenOptions prop on Group

You can pass a prop named `screenOptions` to the `Group` component to configure screens inside the group, where you can specify an object with different options. The options specified in `screenOptions` apply to all of the screens in the group.

Example:

```
<Stack.Navigator>
  <Stack.Group
    screenOptions={{ headerStyle: { backgroundColor: 'papayawhip' } }}
  >
    <Stack.Screen name="Home" component={HomeScreen} />
    <Stack.Screen name="Profile" component={ProfileScreen} />
  </Stack.Group>
  <Stack.Group screenOptions={{ presentation: 'modal' }}>
    <Stack.Screen name="Settings" component={Settings} />
    <Stack.Screen name="Share" component={Share} />
  </Stack.Group>
</Stack.Navigator>
```

Try this example on Snack [↗](#)

Similar to `options`, you can also pass a function to `screenOptions`. The function will receive the `navigation` prop and the `route` prop for each screen. This can be useful if you want to configure options for all the screens in one place based on the route:

```
<Stack.Navigator>
  <Stack.Screen name="Home" component={HomeScreen} />
  <Stack.Screen name="Profile" component={ProfileScreen} />
  <Stack.Group
    screenOptions={({ navigation }) => ({
      presentation: 'modal',
      headerLeft: () => <CancelButton onPress={navigation.goBack} />,
    })}
  >
    <Stack.Screen name="Settings" component={Settings} />
    <Stack.Screen name="Share" component={Share} />
  </Stack.Group>
</Stack.Navigator>
```

screenOptions prop on the navigator

You can pass a prop named `screenOptions` to the navigator component, where you can specify an object with different options. The options specified in `screenOptions` apply to all of the screens in the navigator. So this is a good place to add specify options that you want to configure for the whole navigator.

Example:

```
<Stack.Navigator
  screenOptions={{ headerStyle: { backgroundColor: 'papayawhip' } }}
>
  <Stack.Screen name="Home" component={HomeScreen} />
  <Stack.Screen name="Profile" component={ProfileScreen} />
</Stack.Navigator>
```

Similar to `options`, you can also pass a function to `screenOptions`. The function will receive the `navigation` prop and the `route` prop for each screen. This can be useful if you want to configure options for all the screens in one place based on the route:

```
<Tab.Navigator
  screenOptions={({ route }) => ({
    tabBarIcon: ({ color, size }) => {
      const icons = {
```

```
    Home: 'home',
    Profile: 'account',
  };

  return (
    <MaterialCommunityIcons
      name={icons[route.name]}
      color={color}
      size={size}
    />
  );
},
}}}
>
  <Tab.Screen name="Home" component={HomeScreen} />
  <Tab.Screen name="Profile" component={ProfileScreen} />
</Tab.Navigator>
```

Try this example on Snack [↗](#)

navigation.setOptions method

The `navigation` prop has a `setOptions` method that lets you update the options for a screen from within a component. See [navigation prop's docs](#) more details.

```
<Button
  title="Update options"
  onPress={() => navigation.setOptions({ title: 'Updated!' })}
/>
```

[✎](#) Edit this page