Fundamentals Glossary of terms

**Version: 6.x**

# Glossary of terms

> This is a new section of the documentation and it's missing a lot of terms! Please submit a pull request or an issue with a term that you think should be explained here.

## Navigator

A `Navigator` is React component that decides how to render the screens you have defined. It contains `Screen` elements as its children to define the configuration for screens.

`NavigationContainer` is a component which manages our navigation tree and contains the navigation state. This component must wrap all navigators structure. Usually, we'd render this component at the root of our app, which is usually the component exported from `App.js`.

```
function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator> // <---- This is a Navigator
        <Stack.Screen name="Home" component={HomeScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

## Router

A router is a collection of functions that decide how to handle actions and state changes in the navigator (similar to reducers in Redux apps). Normally you'd never need to interact with a router directly, unless you're writing a custom navigator.

## Screen component

A screen component is a component that we use in our route configuration.

```
const Stack = createNativeStackNavigator();

const StackNavigator = (
  <Stack.Navigator>
    <Stack.Screen
      name="Home"
      component={HomeScreen} // <----
    />
    <Stack.Screen
      name="Details"
      component={DetailsScreen} // <----
    />
  </Stack.Navigator>
);
```

The suffix `Screen` in the component name is entirely optional, but a frequently used convention; we could call it `Michael` and this would work just the same.

We saw earlier that our screen components are provided with the `navigation` prop. It's important to note that *this only happens if the screen is rendered as a route by React Navigation* (for example, in response to `navigation.navigate`). For example, if we render `DetailsScreen` as a child of `HomeScreen`, then `DetailsScreen` won't be provided with the `navigation` prop, and when you press the "Go to Details... again" button on the Home screen, the app will throw one of the quintessential JavaScript exceptions "undefined is not an object".

```
function HomeScreen() {
  return (
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
      <Text>Home Screen</Text>
      <Button
        title="Go to Details"
        onPress={() => navigation.navigate('Details')}
      />
      <DetailsScreen />
    </View>
  );
}
```

The "Navigation prop reference" section goes into more detail on this, describes workarounds, and provides more information on other properties available on `navigation` prop.

## Navigation Prop

This prop will be passed to all screens, and it can be used for the following:

- `dispatch` will send an action up to the router
- `navigate`, `goBack`, etc are available to dispatch actions in a convenient way

Navigators can also accept a navigation prop, which they should get from the parent navigator, if there is one.

For more details, see the "Navigation prop document".

The "Route prop reference" section goes into more detail on this, describes workarounds, and provides more information on other properties available on `route` prop.

## Route Prop

This prop will be passed to all screens. Contains information about current route i.e. `params`, `key` and `name`.

## Navigation State

The state of a navigator generally looks something like this:

```
{
  key: 'StackRouterRoot',
  index: 1,
  routes: [
    { key: 'A', name: 'Home' },
    { key: 'B', name: 'Profile' },
  ]
}
```

For this navigation state, there are two routes (which may be tabs, or cards in a stack). The index indicates the active route, which is "B".

You can read more about the navigation state here.

## Route

Each route is an object which contains a key to identify it, and a "name" to designate the type of route. It can also contain arbitrary params:

```
{
  key: 'B',
  name: 'Profile',
  params: { id: '123' }
}
```

## Header

Also known as navigation header, navigation bar, app bar, and probably many other things. This is the rectangle at the top of your screen that contains the back button and the title for your screen. The entire rectangle is often referred to as the header in React Navigation.

Edit this page