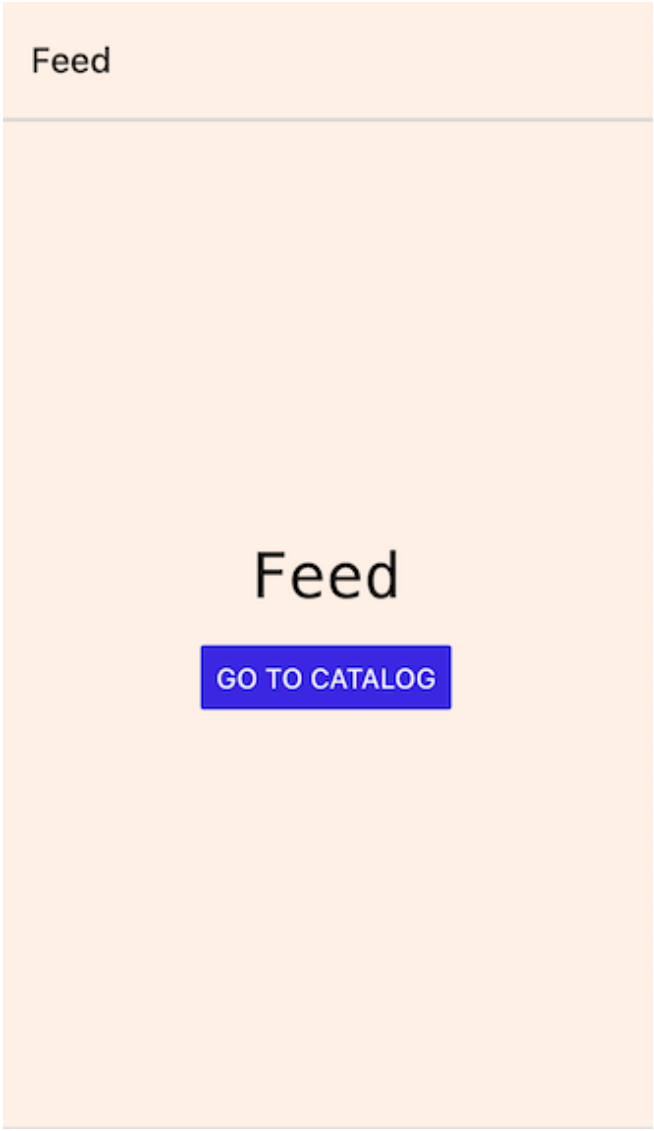
Cheatsheets / **Learn React Native**

# Navigation

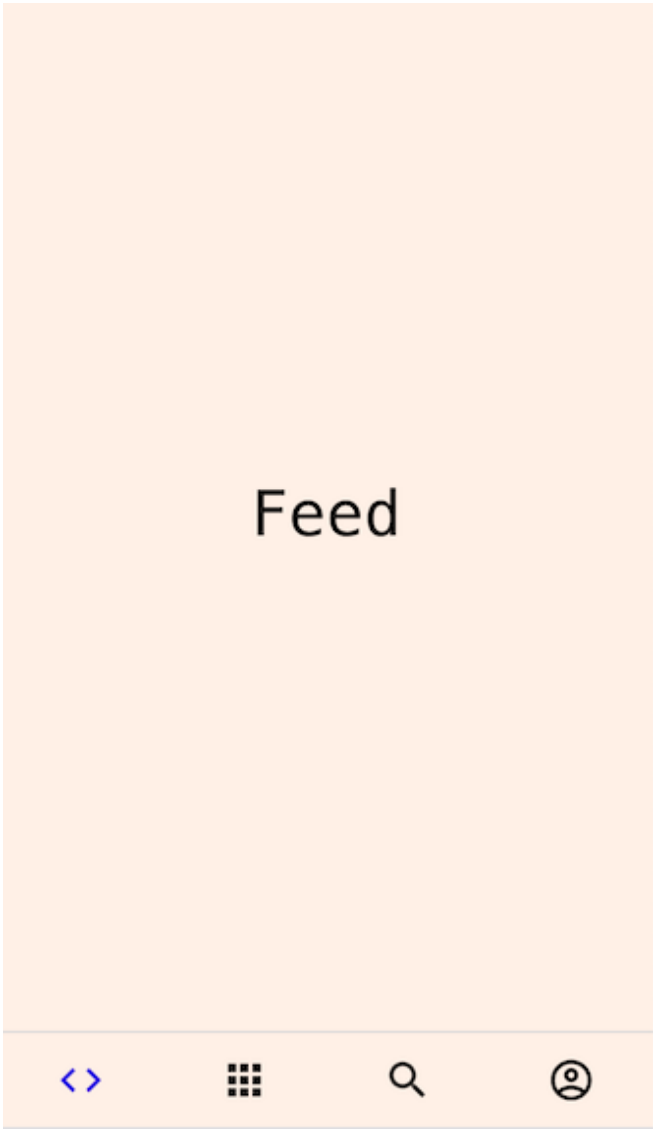
## Stack Navigation

In the *stack navigation* pattern, a user has to go from screen to screen to navigate through all screens, where each one is pushed on a stack. The only UI rendered is a header with the screen title and a back button.



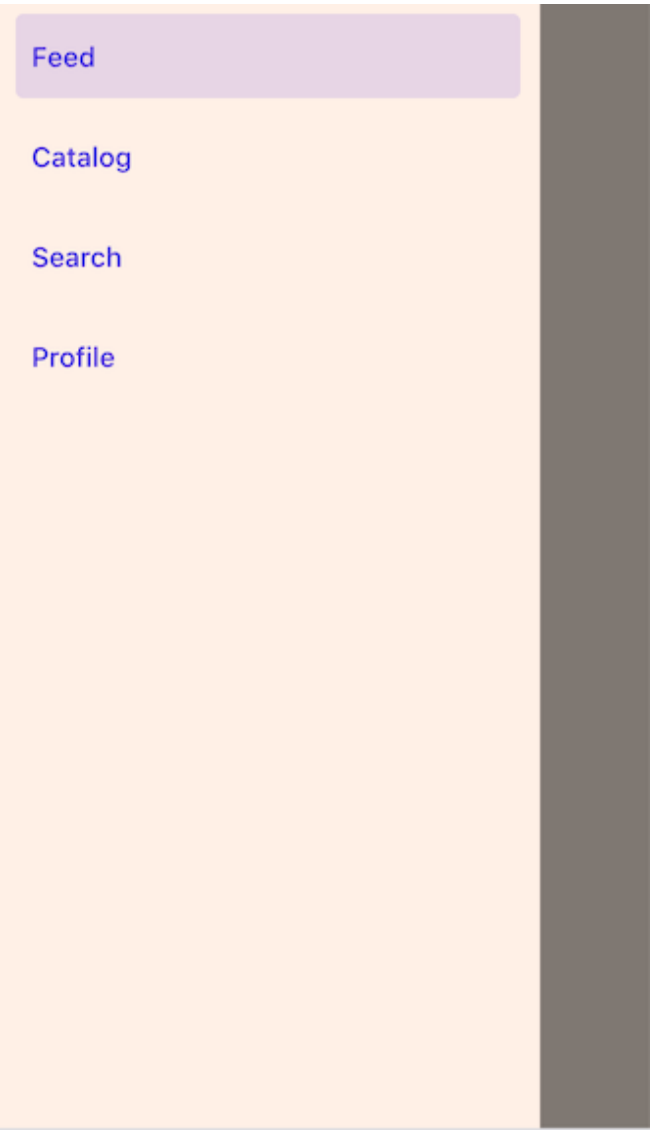
## Tab Navigation

In the *tab navigation* pattern, a user uses a tab bar to switch between screens.



## Drawer Navigation

In the *drawer navigation* pattern, a user uses a pane that can be opened by either swiping or tapping a button, which provides a menu where users can switch between screens.




## `NavigationContainer` Component

In the `react-navigation` library, components to be organized must be wrapped in a `NavigationContainer` component since it keeps track of the navigation structure and makes sure the navigators can operate.

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

const App = () => (
  <NavigationContainer>
        { /* Insert your navigators and
content here */ }
  </NavigationContainer>
);
```

## `createStackNavigator` Factory Method

In the `react-navigation` library, the stack navigator is created by the `createStackNavigator` factory method.

```
const Stack = createStackNavigator();

<Stack.Navigator>
  <Stack.Screen name="Feed" component=
{FeedScreen} />
  <Stack.Screen name="Catalog" component=
{CatalogScreen} />
</Stack.Navigator>
```

## `createBottomTabNavigator` Factory Method

In the `react-navigation` library, the bottom tab navigator is created by the `createBottomTabNavigator` factory method.

```
const Tab = createBottomTabNavigator();

<Tab.Navigator>
  <Tab.Screen name="Feed" component=
{FeedScreen} />
  <Tab.Screen name="Catalog" component=
{CatalogScreen} />
</Tab.Navigator>
```

## useNavigation Hook

In the `react-navigation` library, the `useNavigation` hook provides access to the navigation API and can be used to move users to different screens. It returns an object which is also passed as a `navigation` prop to screens and has multiple methods, including `navigate` (takes a screen name argument) and `goBack`.

```
// Using properties, only available in
screen components
const FeedScreen = (props) => {
  const nav = props.navigation;

  return (
    <Button
      title="Go to home"
      onPress={() =>
nav.navigate('Home')}
    />
  );
};

// Using the hook, available in all
components
const HomeButton = () => {
  const nav = useNavigation();

  return (
    <Button
      title="Go to home"
      onPress={() =>
nav.navigate('Home')}
    />
  );
};
```

## Factory Methods

In the `react-navigation` library, all navigators are created by factory methods that use a naming pattern like `create<type>Navigator()`, which returns an object with `Navigator` and `Screen` properties. These properties are unique components that you must use when rendering the navigation structure.

```
const Stack = createStackNavigator();
const Tab = createBottomTabNavigator();

// You can replace Tab with any other
factory method result.
<Tab.Navigator>
  <Tab.Screen name="Feed" component=
{FeedScreen} />
  <Tab.Screen name="Catalog" component=
{CatalogScreen} />
</Tab.Navigator>
```

Print Share