**Creating and Configuring Native Stack Navigator**

In this reading, you will explore how to create and configure the native stack navigator from React Navigation within the Little Lemon app. Previously you learned how to use Stack Navigator, and in this reading, you will examine the code step by step.

**Install Native Stack Navigator Library**

The very first step in configuring the Native Stack Navigator is to install the library within your React Native app as follows:

1

2

3

npm install @react-navigation/native-stack

This library depends on the **react-native-screens** library you would have installed initially while setting up React Navigation.

**Create Native Stack Navigator**

To create the native stack navigator, you will first import the **createNativeStackNavigator** from the library you just installed **@react-navigation/native-stack***.*

1

import { createNativeStackNavigator } from '@react-navigation/native-stack';

The **createNativeStackNavigator** is a function that returns an object containing two properties. The two properties are **Screen** and **Navigator***.* Both are React components, and you will use them to configure the navigator. You can instantiate the object for it as follows:

1

const Stack = createNativeStackNavigator();

**Imports**

In this example, you will import the two screens, **MenuScreen** and **WelcomeScreen**, which you will be navigating between in the Little Lemon app.

1

2

3

4

5

6

7

import \* as React from 'react';

import { View, Text } from 'react-native';

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

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import MenuScreen from "./Screens/MenuScreen";

import WelcomeScreen from "./Screens/WelcomeScreen";

**Configure Native Stack Navigator**

Remember that the entire app must be wrapped within the **NavigationContainer***,* as seen in the previous sections. Here is how the App component code is written once you have set up the Native Stack Navigator.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

const Stack = createNativeStackNavigator();

function App() {

  return (

    <NavigationContainer>

      <Stack.Navigator initialRouteName="Welcome">

        <Stack.Screen name="Welcome" component={WelcomeScreen} />

        <Stack.Screen name="Menu" component={MenuScreen} />

      </Stack.Navigator>

    </NavigationContainer>

  );

}

export default App;

In this code, the app is wrapped within the **NavigationContainer***.* The **Stack** object contains both the **Navigator** and **Screen** properties. To the **Navigator***,* you can pass the initial route name. Here the initial route has been set to the **Welcome** screen.

The **Screen** component can define routes within your app. And each screen in the stack accepts a **name** prop and a **component** prop. The component is the name of the imported component itself. The **name** prop is displayed as a heading on top of the screen when you navigate to each screen. Notice that the stack now has two routes, a **Welcome** route and a **Menu** route.

By completing this reading, you've learned to create and configure the Native Stack Navigator within your Little Lemon app.

Mark as completed

Like

Dislike

Report an issue