**https://dev.to/easybuoy/combining-stack-tab-drawer-navigations-in-react-native-with-react-navigation-5-da**

**React Navigation Dependencies & Setup**

The dependencies below are the core utility used by the navigators to create the navigation structure, as well as our Stack, Tab, and Drawer navigation.

In your project directory, run the command below on your terminal

**npm install @react-navigation/native @react-navigation/stack @react-navigation/bottom-tabs @react-navigation/drawer**

These dependencies below are the dependency relied upon for gestures, animations, and transitions. Also, run the command below on your terminal to install the dependencies.

npm install react-native-reanimated react-native-gesture-handler react-native-screens react-native-safe-area-context @react-native-community/masked-view

In the root file of our app, in this case, the App.js file, we would set up our react-navigation there.

// ./App.js

import React from "react";

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

const App = () => {

return <NavigationContainer>{/\* Navigation here \*/}</NavigationContainer>;

}

export default App;

Screens setup

The next step would be to set up screens that would be used in our application.

I would be setting up 3 screens for the purpose of this demo, feel free to just copy them.

So I'd set them up in a screens directory. Create a screens directory in the root of the project and add the components below

Home Screen

// ./screens/Home.js

import React from "react";

import { View, Button, Text, StyleSheet } from "react-native";

const Home = () => {

return (

<View style={styles.center}>

<Text>This is the home screen</Text>

<Button title="Go to About Screen" />

</View>

);

};

const styles = StyleSheet.create({

center: {

flex: 1,

justifyContent: "center",

alignItems: "center",

textAlign: "center",

},

});

export default Home;

About Screen

// ./screens/About.js

import React from "react";

import { View, StyleSheet, Text } from "react-native";

const About = () => {

return (

<View style={styles.center}>

<Text>This is the about screen</Text>

</View>

);

};

const styles = StyleSheet.create({

center: {

flex: 1,

justifyContent: "center",

alignItems: "center",

textAlign: "center",

},

});

export default About;

Contact Screen

// ./screens/Contact.js

import React from "react";

import { View, StyleSheet, Text } from "react-native";

const Contact = () => {

return (

<View style={styles.center}>

<Text>This is the contact screen</Text>

</View>

);

};

const styles = StyleSheet.create({

center: {

flex: 1,

justifyContent: "center",

alignItems: "center",

textAlign: "center",

},

});

export default Contact;

Stack Navigation

Stack navigation provides a way for react-native apps to transition between screens by using a stack, which means the screens are stacked on each other.

For example, if you navigate from login to signup screen, the signup screen is stacked on top of the login screen, and if you navigate back, the signup screen is then popped off the stack.

To set up the stack navigation, I'd create a navigation directory at the root of our project. Inside our newly created directory, I'd also create a StackNavigator.js file in there, and add our stack navigation setup.

Note: You can decide to name the folders and files however you want

// ./navigation/StackNavigator.js

import React from "react";

import { createStackNavigator } from "@react-navigation/stack";

import Home from "../screens/Home";

import About from "../screens/About";

const Stack = createStackNavigator();

const MainStackNavigator = () => {

return (

<Stack.Navigator>

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

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

</Stack.Navigator>

);

}

export { MainStackNavigator };

You can also customize and style the stack navigation by adding screenOptions prop style, see basic example below

// ./navigation/StackNavigator.js

const MainStackNavigator = () => {

return (

<Stack.Navigator

screenOptions={{

headerStyle: {

backgroundColor: "#9AC4F8",

},

headerTintColor: "white",

headerBackTitle: "Back",

}}

>

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

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

</Stack.Navigator>

);

}

Back in our App.js file, we can import and add our newly created Stack Navigator. So our App.js file would look like the code below

// ./App.js

import React from "react";

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

import { MainStackNavigator } from "./navigation/StackNavigator";

const App = () => {

return (

<NavigationContainer>

<MainStackNavigator />

</NavigationContainer>

);

}

export default App

Now if we run the code on our emulator, We should now see our Home screen rendering on our Stack screens.

Stack Screen

Remember in our ./screens/Home.js file, we had a button that did nothing, but since we have our stack navigation setup, we can now have access to navigation prop injected by the stack navigation which can help us perform many operations, one of which is redirecting.

So navigate to ./screens/Home.js and add the code below.

// ./screens/Home.js

import React from "react";

import { View, Button, Text, StyleSheet } from "react-native";

const Home = ({ navigation }) => {

return (

<View style={styles.center}>

<Text>This is the home screen</Text>

<Button

title="Go to About Screen"

onPress={() => navigation.navigate("About")} // We added an onPress event which would navigate to the About screen

/>

</View>

);

};

const styles = StyleSheet.create({

center: {

flex: 1,

justifyContent: "center",

alignItems: "center",

textAlign: "center",

},

});

export default Home;

In the code above, we get the navigation prop, which is an object that has a navigate function which we then call passing in the name of the screen we want to navigate to after the button is pressed.

Stack Navigation

And there we have it, we can now navigate between screens using our stack navigation.

Tab Navigation

Tab navigation is a navigation that is tabbed at either the bottom or top of a screen and can be used to switch between different screens.

Tab navigation can take in either the screen as a component or a Stack as the component.

In our StackNavigator.js file, let's create another stack for our contact screen. So our StackNavigator.js will look like below

// ./navigation/StackNavigator.js

import React from "react";

import { createStackNavigator } from "@react-navigation/stack";

import Home from "../screens/Home";

import About from "../screens/About";

import Contact from "../screens/Contact";

const Stack = createStackNavigator();

const screenOptionStyle = {

headerStyle: {

backgroundColor: "#9AC4F8",

},

headerTintColor: "white",

headerBackTitle: "Back",

};

const MainStackNavigator = () => {

return (

<Stack.Navigator screenOptions={screenOptionStyle}>

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

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

</Stack.Navigator>

);

}

const ContactStackNavigator = () => {

return (

<Stack.Navigator screenOptions={screenOptionStyle}>

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

</Stack.Navigator>

);

}

export { MainStackNavigator, ContactStackNavigator };

We can then create another file TabNavigator.js in the navigations directory and add the markup for our TabNavigator

// ./navigation/TabNavigator.js

import React from "react";

import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";

import { MainStackNavigator, ContactStackNavigator } from "./StackNavigator";

const Tab = createBottomTabNavigator();

const BottomTabNavigator = () => {

return (

<Tab.Navigator>

<Tab.Screen name="Home" component={MainStackNavigator} />

<Tab.Screen name="Contact" component={ContactStackNavigator} />

</Tab.Navigator>

);

};

export default BottomTabNavigator;

And then back in our App.js file, we can now import our newly created TabNavigator and use it there.

// ./App.js

import React from "react";

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

import BottomTabNavigator from "./navigation/TabNavigator";

const App = () => {

return (

<NavigationContainer>

<BottomTabNavigator />

</NavigationContainer>

);

}

export default App

Tab Navigation

Drawer Navigation

Drawer navigation is a slide-out and slide-in drawer that contains links to various screens. The Drawer navigation opens when a menu icon is clicked or when a user swipes their finger from the left or right edge of the app.

In order to create drawer navigation, we would create another file in our navigations directory called DrawerNavigator.js

In that file, we would add our drawer navigator markup

// ./navigation/DrawerNavigator.js

import React from "react";

import { createDrawerNavigator } from "@react-navigation/drawer";

import { ContactStackNavigator } from "./StackNavigator";

import TabNavigator from "./TabNavigator";

const Drawer = createDrawerNavigator();

const DrawerNavigator = () => {

return (

<Drawer.Navigator>

<Drawer.Screen name="Home" component={TabNavigator} />

<Drawer.Screen name="Contact" component={ContactStackNavigator} />

</Drawer.Navigator>

);

}

export default DrawerNavigator;

And then back in our App.js file, we can now import our newly created DrawerNavigtor.js file and use it as our navigator.

// ./App.js

import React from "react";

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

import DrawerNavigator from "./navigation/DrawerNavigator";

const App = () => {

return (

<NavigationContainer>

<DrawerNavigator />

</NavigationContainer>

);

}

export default App;