

1. Drawer Navigator (เมนูด้านข้างแบบเลื่อน)

Drawer Navigator เป็นการนำทางแบบ “เลื่อนจากด้านข้าง” ซึ่งเหมาะสำหรับเมนูหลักหรือเมนูที่ไม่ต้องให้เห็นตลอดเวลา เช่น Settings, Profile, Logout

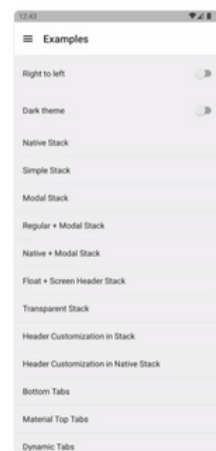
```
import * as React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createDrawerNavigator } from '@react-navigation/drawer';
import { View, Text } from 'react-native';

function HomeScreen() {
  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <Text>หน้าหลัก</Text>
    </View>
  );
}

function ProfileScreen() {
  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <Text>โปรไฟล์</Text>
    </View>
  );
}

const Drawer = createDrawerNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Drawer.Navigator initialRouteName="Home">
        <Drawer.Screen name="Home" component={HomeScreen} />
        <Drawer.Screen name="Profile" component={ProfileScreen} />
      </Drawer.Navigator>
    </NavigationContainer>
  );
}
```



สร้างหน้าจอ (Screens):

- HomeScreen และ ProfileScreen คือสองหน้าจอที่เราจะแสดงใน Drawer Navigator

การสร้าง Drawer Navigator:

- ใช้ createDrawerNavigator() เพื่อสร้างตัว Navigator สำหรับ Drawer
- initialRouteName="Home" กำหนดว่าเมื่อเริ่มแอปจะเปิดหน้าจอ Home ก่อน

การกำหนดเมนูใน Drawer:

- ใช้ Drawer.Screen เพื่อกำหนดหน้าจอที่จะแสดงในเมนู (ในที่นี้มี Home และ Profile)

คำอธิบาย

- initialRouteName: ใช้สำหรับกำหนดหน้าจอที่จะแสดงเป็นหน้าจอแรกเมื่อเปิดแอป
- Drawer.Navigator: ใช้เพื่อสร้างแถบเมนูที่สามารถเลื่อนออกจากข้าง
- Drawer.Screen: ใช้กำหนดหน้าจอที่จะแสดงใน Drawer
- NavigationContainer: เป็นตัวห่อหุ้มแอปทั้งหมดที่ใช้ React Navigation เพื่อให้ระบบการนำทางทำงานได้

สามารถปรับแต่งได้หลายอย่าง เช่น สีของ Drawer, ตัวเลือกการวางเมนู หรือการใช้งาน Custom Drawer โดยใช้ drawerStyle หรือ drawerContent

```
<Drawer.Navigator
  drawerStyle={{
    backgroundColor: '#f0f0f0', // สีพื้นหลังของ Drawer
    width: 240, // กำหนดความกว้างของ Drawer
  }}
  drawerContent={() => (
    <CustomDrawerContent {...props} /> // ใช้ custom component
  )}
>
  <Drawer.Screen name="Home" component={HomeScreen} />
  <Drawer.Screen name="Profile" component={ProfileScreen} />
</Drawer.Navigator>
```

2. Bottom Tabs Navigator (เมนูด้านล่าง)

Bottom Tabs Navigator เป็นการนำทางที่แสดงเมนูไว้ด้านล่าง เหมาะกับแอปที่ต้องการเข้าถึงหน้าหลัก ๆ อย่างรวดเร็ว เช่น Home, Search, Settings

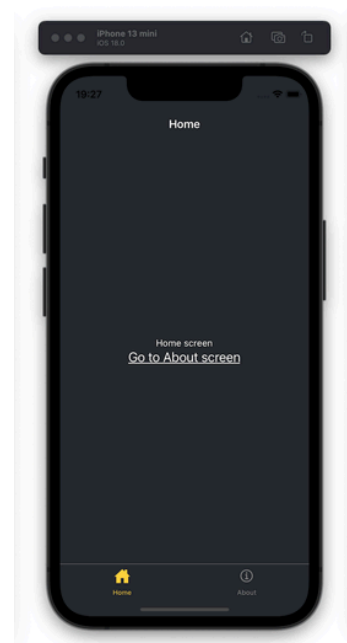
```
import * as React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
import { View, Text } from 'react-native';
import { Ionicons } from '@expo/vector-icons'; // สำหรับไอคอน

function HomeScreen() {
  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <Text>หน้าหลัก</Text>
    </View>
  );
}

function SettingsScreen() {
  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <Text>ตั้งค่า</Text>
    </View>
  );
}

const Tab = createBottomTabNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Tab.Navigator
        screenOptions={({ route }) => ({
          tabBarIcon: ({ color, size }) => {
            let iconName;
            if (route.name === 'Home') {
              iconName = 'home';
            } else if (route.name === 'Settings') {
              iconName = 'settings';
            }
            return <Ionicons name={iconName} size={size} color={color} />;
          },
        })}
      >
        <Tab.Screen name="Home" component={HomeScreen} />
        <Tab.Screen name="Settings" component={SettingsScreen} />
      </Tab.Navigator>
    </NavigationContainer>
  );
}
```



สร้างหน้าจอ (Screens):

- HomeScreen และ SettingsScreen คือสองหน้าจอที่เราจะแสดงบน Bottom Tab

การสร้าง Navigator:

- ใช้ createBottomTabNavigator() เพื่อสร้างตัว Navigator สำหรับ Bottom Tabs

การแสดงไอคอนในแต่ละแท็บ:

- ใน screenOptions ใช้ tabBarIcon เพื่อกำหนดไอคอนที่จะแสดงในแต่ละแท็บ โดยใช้ไลบรารี Ionicons สำหรับไอคอน
- สำหรับแท็บ Home ใช้ไอคอน home และสำหรับ Settings ใช้ไอคอน settings

การตั้งค่า NavigationContainer:

- ใช้ NavigationContainer เพื่อห่อหุ้ม Navigator ทั้งหมด (ที่ใช้ใน React Navigation)

เพิ่มเติม

- tabBarIcon: เป็นฟังก์ชันที่กำหนดไอคอนของแท็บ โดยสามารถใช้ไอคอนจาก Ionicons หรือไอคอนอื่น ๆ ตามที่ต้องการ
- screenOptions: ใช้กำหนดตัวเลือกต่าง ๆ สำหรับหน้าจอในแต่ละแท็บ เช่น ไอคอน, สี, และการตั้งค่าอื่น ๆ
- NavigationContainer: จำเป็นต้องใช้ห่อหุ้มแอปทั้งหมดเพื่อให้ React Navigation ทำงานได้

ต้องการให้ไอคอนมีสีที่แตกต่างกันเมื่อแท็บนั้นถูกเลือกหรือไม่ถูกเลือก (เช่น ไอคอนสีน้ำเงินเมื่อเลือกแล้ว, สีเทาเมื่อไม่เลือก) สามารถใช้ `tabBarActiveTintColor` และ `tabBarInactiveTintColor` ได้:

```
<Tab.Navigator
  screenOptions={{
    tabBarActiveTintColor: '#1e90ff',
    tabBarInactiveTintColor: '#d3d3d3',
  }}
>
  <Tab.Screen name="Home" component={HomeScreen} />
  <Tab.Screen name="Settings" component={SettingsScreen} />
</Tab.Navigator>
```