

React Native Navigation: Stack, Tab & Drawer for Beginners

Subtitle: Learn how to navigate between screens and build multi-page mobile apps efficiently.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

Introduction

Navigation is essential for building multi-screen apps in React Native. React Navigation is the most popular library, offering **Stack, Tab, and Drawer navigators** to structure your app. This guide explains each type and provides practical examples.

Step 1: Installing React Navigation

1. Install core package:

```
npm install @react-navigation/native
```

2. Install dependencies:

```
npm install react-native-screens react-native-safe-area-context
```

3. Install navigator packages (example: Stack Navigator):

```
npm install @react-navigation/native-stack
```

4. Wrap your app with `NavigationContainer` from React Navigation.
-

Step 2: Stack Navigation

Stack navigation allows **screen-to-screen transitions** like a call stack.

Example:

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

```

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

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

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

const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={HomeScreen} />
        <Stack.Screen name="Details" component={DetailsScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}

```

Exercise: Add a third screen called **Profile** and navigate to it from the Details screen.

Step 3: Tab Navigation

Tab navigation provides **bottom or top tabs** for easy screen switching.

Example:

```
npm install @react-navigation/bottom-tabs
```

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

```

import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
import HomeScreen from './HomeScreen';
import ProfileScreen from './ProfileScreen';

const Tab = createBottomTabNavigator();

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

```

Exercise: Customize tab icons and labels using the `screenOptions` prop.

Step 4: Drawer Navigation

Drawer navigation lets users **swipe from the side** to access screens.

Example:

```
npm install @react-navigation/drawer
```

```

import { createDrawerNavigator } from '@react-navigation/drawer';
import HomeScreen from './HomeScreen';
import SettingsScreen from './SettingsScreen';

const Drawer = createDrawerNavigator();

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

```

Exercise: Add a **Profile** screen to the drawer and reorder the items.

Step 5: Best Practices

- Use **Stack navigation** for sequential flows.
 - Use **Tab navigation** for main sections.
 - Use **Drawer navigation** for secondary or less-used screens.
 - Combine navigators to build **nested navigation** (e.g., Stack inside Tab).
 - Always test **navigation gestures** on both iOS and Android.
-

Navigation is the backbone of multi-screen apps. Mastering Stack, Tab, and Drawer navigators lets you build **complex yet user-friendly mobile applications**.

Visit **haas.dev** for more React Native guides, tutorials, and projects.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>
