StackNavigator cung cấp cho bạn một cách chuyển qua lại giữa các màn hình với nhau và có thể quản lý nó một cách dễ dàng. Ngoài ra nó còn có thể cung cấp các gestures và animation phù hợp với Android và iOS khi chuyển đổi giữa các màn hình.

**Tạo StackNavigator**

Stacknavigator là một hàm (function) trả về một React component. Nó có thể nhận vào một đối tượng route configuration đã được định nghĩa từ trước. Do StackNavigator trả về một component nên chúng ta có thể export nó trực tiếp trong file App.js và coi nó như một root component.

// Trong file App.js

\'use strict\';

import React from \'react\';

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

import { StackNavigator } from \'react-navigation\';

class HomeScreen extends React.Component {

render() {

return (

<View style={{ flex: 1, alignItems: \'center\', justifyContent: \'center\' }}>

<Text>Home Screen</Text>

</View>

);

}

}

export default StackNavigator({

Home: {

screen: HomeScreen,

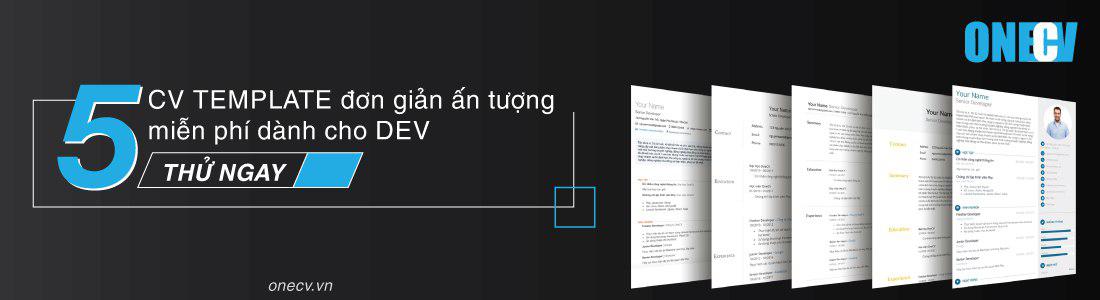
},

});

Chạy đoạn code trên, bạn sẽ thấy màn hình với một navigation bar và chữ Home Screen ở giữa màn hình. Có thể hiểu là chúng ta đã tạo ra được một StackNavigator và StackNavigator này chứa màn hình HomeScreen. Mới đến đây thì chưa thấy được tác dụng của StackNavigator ngoài việc có thêm một thanh navigation bar cả.

**Chuyển đổi giữa các màn hình**

Chúng ta sẽ tạo thêm một màn hình nữa rồi sau đó sẽ chuyển đổi giữa các màn hình với nhau:

[](https://onecv.vn/)

// App.js

class HomeScreen extends React.Component {

render() {

return (

<View style={{ flex: 1, alignItems: \'center\', justifyContent: \'center\' }}>

<Text>Home Screen</Text>

</View>

);

}

}

class DetailsScreen extends React.Component {

render() {

return (

<View style={{ flex: 1, alignItems: \'center\', justifyContent: \'center\' }}>

<Text>Details Screen</Text>

</View>

);

}

}

export default StackNavigator({

Home: {

screen: HomeScreen,

},

Details: {

screen: DetailsScreen,

},

});

Khi chạy lại app thì vẫn chỉ thấy mỗi màn hình Home hiện ra, tiếp tục thêm một Button trong màn hình Home để có thể chuyển qua màn hình DetailsScreen. Sửa code trong màn hình Home như sau:

\'use strict\';

import React from \'react\';

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

import { StackNavigator } from \'react-navigation\';

class HomeScreen extends React.Component {

render() {

return (

<View style={{ flex: 1, alignItems: \'center\', justifyContent: \'center\' }}>

<Text>Home Screen</Text>

<Button

title="Go to Details"

onPress={() => this.props.navigation.navigate(\'Details\')}

/>

</View>

);

}

}

Lưu ý:

* this.props.navigation: navigation prop được tạo ra ở tất cả các component được định nghĩa trong StackNavigator
* navigate(\'Details\'): đây là hàm dùng để chuyển đổi tới một màn hình khác có trong cùng một StackNavigator.

Ok, chạy lại app và ấn vào button Go to Details thì đã chuyển được sang màn hình DetailsScreen. Để ý thấy navigation bar cũng có sự thay đổi là có thêm nút back để chúng ta có thể quay lại màn hình trước đó. Có thể hiểu rằng StackNavigator đã cho HomeScreen và DetailsScreen vào một chỗ để có thể quản lý luồng của 2 màn hình này.

Chúng ta có thể thay đổi một chút để trông code dễ đọc hơn như sau:

// App.js

\'use strict\';

import React from \'react\';

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

import { StackNavigator } from \'react-navigation\';

class HomeScreen extends React.Component {

render() {

return (

<View style={{ flex: 1, alignItems: \'center\', justifyContent: \'center\' }}>

<Text>Home Screen</Text>

<Button

title="Go to Details"

onPress={() => this.props.navigation.navigate(\'Details\')}

/>

</View>

);

}

}

class DetailsScreen extends React.Component {

render() {

return (

<View style={{ flex: 1, alignItems: \'center\', justifyContent: \'center\' }}>

<Text>Details Screen</Text>

</View>

);

}

}

const RootStack = StackNavigator({

Home: {

screen: HomeScreen,

},

Details: {

screen: DetailsScreen,

},

});

export default class App extends React.Component {

render() {

return <RootStack />;

}

}