



React Native cơ bản

Bài 6: Màn hình Chính và Thực đơn

<http://csc.edu.vn/laptrinh/>

Màn hình chính và thực đơn



- ☐ React-navigation
- ☐ Màn hình chính
- ☐ Minh họa

- ❑ Giới thiệu
- ❑ Điều hướng và truyền tham số

- Là các component mở rộng, hỗ trợ việc điều hướng và truyền tham số giữa các màn hình.
- Cung cấp các phương thức hỗ trợ tạo thực đơn dạng Tab và Slider.

Điều hướng và truyền tham số



- Là việc chuyển từ màn hình này sang màn hình khác (hay ngược lại). Trong đó cho phép truyền tham số giữa các màn hình

Cài đặt và cấu hình React-navigation



- Cài đặt
 - `npm i react-navigation`
 - `npm i react-native-gesture-handler`
 - `npm i react-native-reanimated`
 - `npm i react-router-native --s`
 - `npm i react-native-button`
 - `npm i react-navigation-stack`
 - `npm i react-navigation-tabs`
 - `npm i react-navigation-drawer`

Cài đặt và cấu hình React-navigation



- Cấu hình (android)

- android/app/src/main/java/com/demo/MainActivity.java

```
import com.facebook.react.ReactActivityDelegate;
import com.facebook.react.ReactRootView;
import com.swmansion.gesturehandler.react.RNGestureHandlerEnabledRootView;
public class MainActivity extends ReactActivity {
    @Override
    protected String getMainComponentName() {
        return "man_hinh_chinh";
    }
    @Override
    protected ReactActivityDelegate createReactActivityDelegate() {
        return new ReactActivityDelegate(this, getMainComponentName()) {
            @Override
            protected ReactRootView createRootView() {
                return new RNGestureHandlerEnabledRootView(MainActivity.this);
            }
        };
    }
}
```



- Xây dựng Màn hình với react-router-native

```
import React, { Component } from 'react';  
import { Text, View, StyleSheet } from 'react-native';  
import { NativeRouter, Route, Link } from 'react-router-native'
```

```
const Home = () => <Text style={styles.header}>Home</Text>;  
const About = () => <Text style={styles.header}>About</Text>;  
const Topic = ({ match }) => (<Text style={styles.topic}>{match.params.topicId}</Text>);
```


- Xây dựng Màn hình với react-router-native

```
const Topics = ({ match }) => (  
  <View>  
    <Text style={styles.header}>Topics</Text>  
    <View>  
      <Link to={`${match.url}/rendering`} style={styles.subNavItem} underlineColor="#f0f4f7">  
        <Text>Rendering with React</Text>  
      </Link>  
      <Link to={`${match.url}/components`} style={styles.subNavItem} underlineColor="#f0f4f7">  
        <Text>Components</Text>  
      </Link>  
      <Link to={`${match.url}/props-v-state`} style={styles.subNavItem} underlineColor="#f0f4f7">  
        <Text>Props v. State</Text>  
      </Link>  
    </View>  
    <Route path={`${match.path}/:topicId`} component={Topic} />  
    <Route exact path={match.path} render={() => <Text style={styles.topic}>Please select a topic.</Text>} />  
  </View>  

```






- Xây dựng Màn hình với react-router-native

```
const navigation = () => (  
  <NativeRouter>  
    <View style={styles.container}>  
      <View style={styles.nav}>  
        <Link to="/" underlayColor="#f0f4f7" style={styles.navItem}>  
          <Text>Home</Text>  
        </Link>  
        <Link to="/about" underlayColor="#f0f4f7" style={styles.navItem}>  
          <Text>About</Text>  
        </Link>  
        <Link to="/topics" underlayColor="#f0f4f7" style={styles.navItem}>  
          <Text>Topics</Text>  
        </Link>  
      </View>  
  
      <Route exact path="/" component={Home} />  
      <Route path="/about" component={About} />  
      <Route path="/topics" component={Topics} />  
    </View>  
  </NativeRouter>  
)
```


- Xây dựng Màn hình với react-router-native


```
const styles = StyleSheet.create({
  container: { marginTop: 25,padding: 10},
  header: { fontSize: 20},
  nav: {
    flexDirection: "row",
    justifyContent: "space-around"
  },
  navItem: {
    flex: 1,
    alignItems: "center",
    padding: 10
  },
  subNavItem: {
    padding: 5
  },
  topic: {
    textAlign: "center",
    fontSize: 15
  }
});
export default navigation
```


- Xây dựng chức năng xem danh sách Tivi.

Danh sách Tivi	
	Smart Tivi Cong Samsung 49 inch Đơn giá: 22,900,000 VND
	Internet Tivi Sony 4K 43 inch Đơn giá: 14,500,000 VND
	Internet Tivi Samsung 32 inch Đơn giá: 6,890,000 VND
	Internet Tivi Sony 40 inch Đơn giá: 9,990,000 VND
	Smart Tivi Samsung 40 inch Đơn giá: 8,640,000 VND

Ví dụ

- Khi chọn 1 Tivi từ danh sách chuyển sang chức năng xem thông tin Tivi tương ứng
- Khi chọn  trở về chức năng xem danh sách Tivi

 Internet Tivi Sony 4K 43 inch



- Tham chiếu đối tượng StackNavigator:
 - `import { createStackNavigator } from 'react-navigation-stack';`
 - `import { createAppContainer } from 'react-navigation';`
- Khởi tạo đối tượng StackNavigator:

```
const Man_hinh = createStackNavigator({  
  "Man_hinh_Danh_sach_Tivi": {  
    screen: Danh_sach_Tivi  
  },  
  "Man_hinh_Thong_tin_Tivi": {  
    screen: Thong_tin_Tivi  
  }  
});
```

- Khởi tạo đối tượng AppContainer

```
const App = createAppContainer(Man_hinh);  
export default App
```

- Phương thức navigate: điều hướng, cho phép truyền tham số

```
XL_Chon(){  
  |   this.props.navigation.navigate("Man_hinh_Thong_tin_Tivi",{Tivi:this.props.item})  
  | }  
}
```

- Nhận trị từ tham số truyền

```
constructor(props){  
  |   super(props);  
  |   let Tivi= this.props.navigation.state.params.Tivi;  
  |   this.state={  
  |     |   Ma_so:Tivi.Ma_so,  
  |     |   Ten:Tivi.Ten,  
  |     |   Hinh:Tivi.Hinh,  
  |     |   Don_gia:Tivi.Don_gia  
  |   }  
  | }  
}
```

- Phương thức goBack: trở về màn hình trước

```
<TouchableOpacity onPress={this.XL_Tro_ve.bind(this)}>  
  |   <Image style={style.inputIcon} source={require('../images/back.png')}/>  
  | </TouchableOpacity>
```

- ☐ Giới thiệu
- ☐ Thực đơn dạng Tab
- ☐ Thực đơn dạng Slider

- Là màn hình chứa thực đơn (tên danh sách các chức năng) liên kết đến các chức năng tương ứng trong ứng dụng
- Thường được hiển thị sau khi ứng dụng khởi động xong hay sau khi người dùng đăng nhập thành công.

Thực đơn dạng Tab



- Là danh sách các chức năng mà tại một thời điểm chỉ hiển thị nội dung của một chức năng, chúng ta có thể chọn chức năng khác để hiển thị nội dung tương ứng.

- Xây dựng Màn hình chính với thực đơn dạng Tab:



- Khi chọn chức năng trong thực đơn sẽ hiển thị chức năng tương ứng



- Tham chiếu đối tượng TabNavigator:
 - import { createBottomTabNavigator } from 'react-navigation-tabs';
 - import { createAppContainer } from 'react-navigation';
- Khởi tạo đối tượng TabNavigator :

```
const routeConfigs = {  
  Man_hinh_Chinh:{  
    | screen:Chinh  
  },  
  Man_hinh_Gio_hang:{  
    | screen:Gio_hang  
  },  
  Man_hinh_Tim_kiem:{  
    | screen:Tim_kiem  
  },  
  Man_hinh_Lien_he:{  
    | screen:Lien_he  
  },  
};
```

- Cấu hình TabNavigator và Khởi tạo đối tượng BottomTab:

```
const tabNavigatorConfig = {  
  initialRouteName: "Man_hinh_Lien_he",  
  animationEnabled: true,  
  swipeEnabled: true,  
  tabBarOptions: {  
    activeTintColor: 'tomato',  
    inactiveTintColor: 'gray',  
  },  
};  
  
const Man_hinh= createBottomTabNavigator(routeConfigs, tabNavigatorConfig);  
  
const App=createAppContainer(Man_hinh);  
export default App
```

- Sử dụng navigationOptions để định dạng biểu tượng và nội dung hiển thị

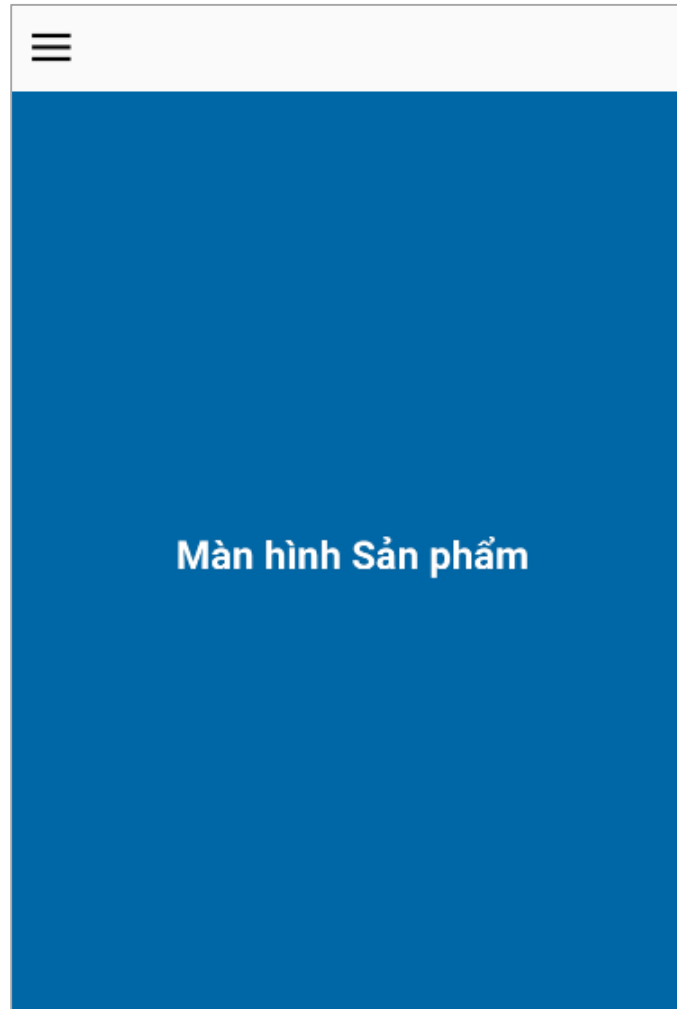
```
static navigationOptions = ({navigation}) =>{
  const {params = {}} = navigation.state;
  let tabBarLabel= "Liên hệ";
  let tabBarIcon=()=>(
    <Image source={require('../icons/customer.png')} style={{width:26, height:26, tintColor:backgroundColor}}/>
  );
  return {tabBarLabel, tabBarIcon};
}
```

Thực đơn dạng Slider

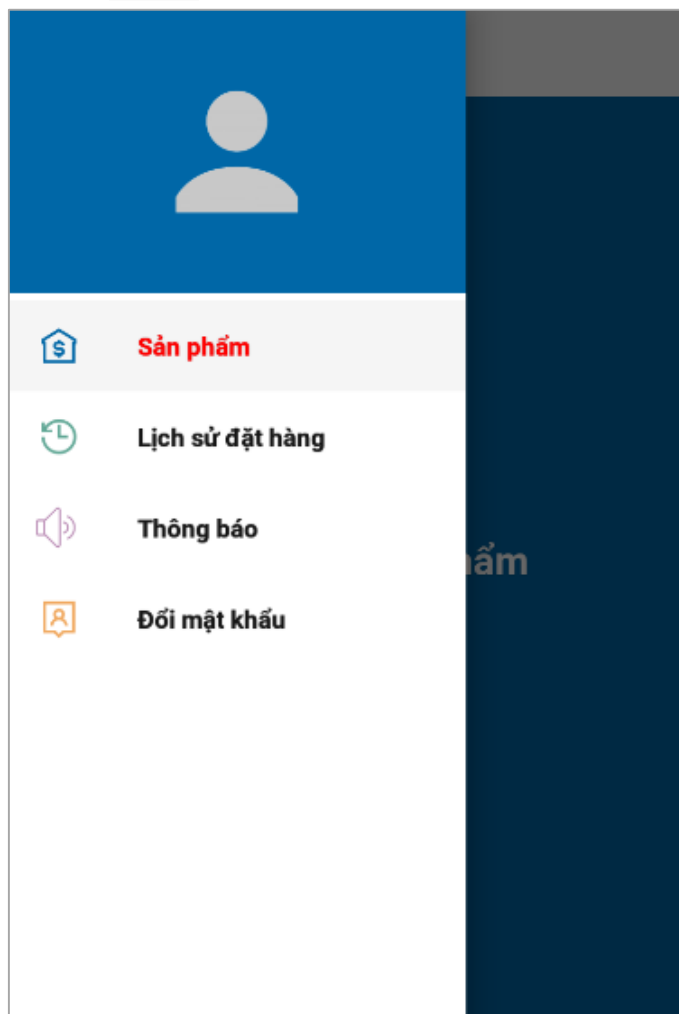


- Là danh sách các chức năng ẩn, được hiển thị khi chúng ta kích hoạt (có thể từ biểu tượng hay slider từ cạnh màn hình)

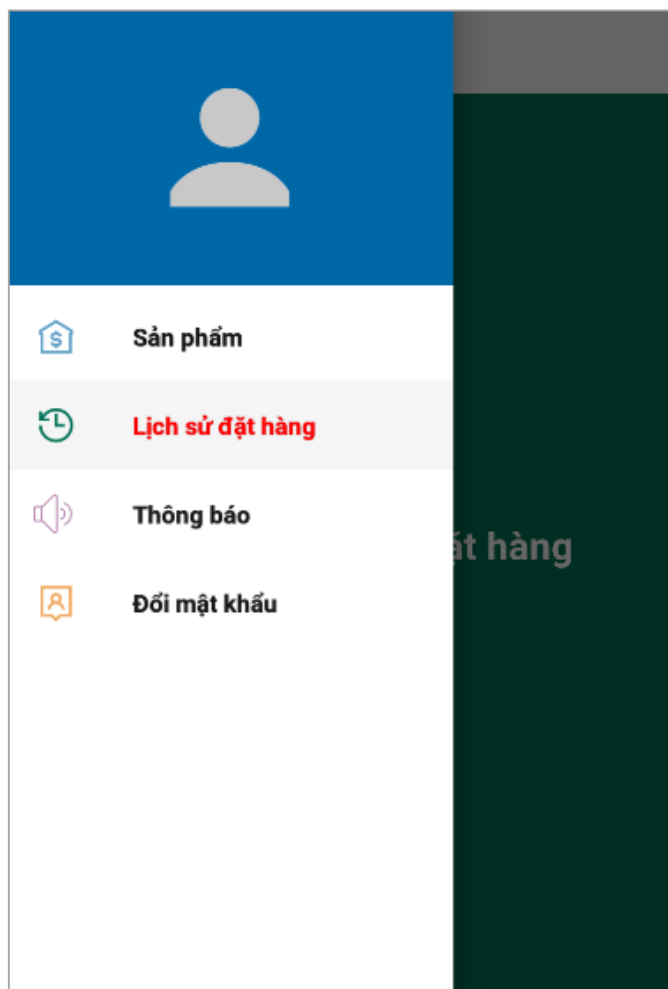
- Xây dựng Màn hình chính với thực đơn dạng Slider:



- Khi chọn biểu tượng  hiển thị thực đơn



- Khi chọn một chức năng, hiển thị màn hình tương ứng:



- Tham chiếu đối tượng DrawerNavigator:
 - import {createDrawerNavigator } from 'react-navigation-drawer';
 - import {DrawerNavigatorItems } from 'react-navigation-drawer'
 - import {createAppContainer} from 'react-navigation';
- Khởi tạo đối tượng DrawerNavigator:

```
const XL_The_hien= (props) => (  
  <View style={style.container}>  
    <View style={{height:160, alignItems:"center", justifyContent:"center",backgroundColor:'#0067a7'}}>  
      <Image style={style.image} source={require('./images/user.png')}></Image>  
    </View>  
    <View style={{flex:1, justifyContent:"flex-start"}}>  
      <DrawerNavigatorItems {...props}/>  
    </View>  
  </View>  
>);
```

Hướng dẫn



```
const routeConfigs = {
  Man_hinh_San_pham:{
    screen:San_pham
  },
  Man_hinh_Lich_su_Dat_hang:{
    screen:Lich_su_Dat_hang
  },
  Man_hinh_Thong_bao:{
    screen:Thong_bao
  },
  Man_hinh_Doi_Mat_khau:{
    screen:Doi_Mat_khau
  },
};

const drawerNavigatorConfig = {
  initialRouteName:"Man_hinh_San_pham",
  drawerWidth: width*2/3,
  drawerPosition:"left",
  contentComponent:XL_The_hien,
  contentOptions:{
    activeTintColor:"red",
  }
};

const Thuc_don_Slider= createDrawerNavigator(routeConfigs, drawerNavigatorConfig);
const App=createAppContainer(Thuc_don_Slider);
const style= StyleSheet.create({
  container:{
    flex:1,
    justifyContent:'center'
  },
  image:{
    width: 80, height: 80,
    borderTopLeftRadius:10,
    borderTopRightRadius:10,
    borderBottomLeftRadius:10,
    borderBottomRightRadius:10
  }
})
export default Man_hinh
```

- Sử dụng navigationOptions để định dạng biểu tượng và nội dung hiển thị

```
static navigationOptions = ({navigation}) =>{
  let drawerLabel= "Sản phẩm";
  let drawerIcon=()=>{
    <Image
      source={require('../icons/rent.png')}
      style={{width:26, height:26, tintColor:backgroundColor}}/>
    );
  return {drawerLabel, drawerIcon};
}
```

- Phương thức navigate("DrawerOpen"): mở thực đơn

```
<View style={{
  flexDirection:"row", height:50,
  justifyContent:'flex-start', alignItems:'center'}}>
  <TouchableHighlight
    style={{marginLeft:10}}
    onPress={()=>{this.props.navigation.openDrawer()}}>
    <Image source={require("../icons/menu.png")} style={{width:26, height:26}}/>
  </TouchableHighlight>
</View>
```

- Giáo viên minh họa