# 页面跳转

在android原生跳转是使用Intent配合startActivity（intent）进行页面跳转的。但是到了React native是使用导航器来完成页面跳转的。

## 简介

React Native主推的导航器是一个单独导航库react-navigation。

提示：

从0.44版本开始，Navigator被从react native的核心组件库中剥离到了一个名为react-native-deprecated-custom-components的单独模块中。如果你需要继续使用Navigator，则需要先npm i facebookarchive/react-native-custom-components安装，然后从这个模块中import，即import { Navigator } from 'react-native-deprecated-custom-components'.

官方的东西就是好东西。

React-navigation中主要是三个组件：

* **StackNavigator**导航组件
* **TabNavigator**切换组件
* **DrawerNavigator**抽屉组件、

StackNavigator 用于实现各个页面之间的跳转， TabNavigator 用来实现同一个页面上不同界面的切换， DrawerNavigator 可以实现侧滑的抽屉效果。

**什么是场景**

场景   通俗的讲就是整个屏幕的排版布局显示，比如 又几个TextView  或者ScrollView等组件组成的一个界面 就是一个场景  或者是一个全屏的React组件。

相对于场景来说 组件就是场景的部分 一般情况下 我们自定义的组件 并不是单纯的在一个场景中使用 很多时候我们需要服用这个组件 那么这个组件就要单独拿出来写成一个js文件

接下来使用的是StackNavigator 进行场景的切换。

## 使用

StackNavigator 放置在一个入口js文件中，在项目初始化后，会存在一个index.js文件，StackNavigator 定义在AppRegistry注册组件的上方。

**const** SimpleApp = StackNavigator({  
 Home: {screen: Login},  
 APP:{screen:App},  
});  
AppRegistry.registerComponent('ReactNativeApp', () => SimpleApp);

registerComponent方法，将界面内容交给simpleApp来实现，实际上StackNavigator 内部存在一个栈用来存放多个场景，Home APP就是其中两个场景,后面的screen指场景对应的显示组件类。Home是第一个入栈的场景，默认作为第一个显示场景，其余的都是自己定义的名称，可随意定义名称。

在组件实例化后，navigation作为属性传递给给个场景，可以通过：

**const** {navigate} = **this**.props.navigation;

获取其实例对象。

如何跳转：

navigate("场景名称");

场景名称已经在StackNavigator 中定义，直接填写相关名称就可以跳转相关页面。

## 举个例子

创建Login.js作为打开应用的第一个页面：

**import** React, {Component} **from** 'react';  
**import** {  
 Text,  
 View,  
 TextInput, TouchableOpacity  
} **from** 'react-native';  
  
**export default class** Login **extends** Component {  
 user = "";  
 pass = "";  
 \_onRegister() {  
 }  
 \_onLogin() {  
 **const** {navigate} = **this**.props.navigation;  
 **if** (**this**.user == "123" && **this**.pass == "123") {  
 navigate("APP");  
 }  
 }  
  
 constructor(props) {  
 **super**(props);  
 **this**.state = {  
 username: "",  
 password: "",  
 }  
 }  
 **static** *navigationOptions* = {  
 title: 'login',  
 };  
 render() {  
 **return** (  
 <View>  
 <View style={{flexDirection: 'column'}}>  
 <TextInput  
 style={{fontSize: 20}}  
 placeholder="username"  
 onChangeText={(username) => **this**.user = username}  
 />  
 <TextInput  
 style={{fontSize: 20}}  
 placeholder="password"  
 onChangeText={(password) => **this**.pass = password}  
 />  
 </View>  
 <View style={{flexDirection: 'row', justifyContent: 'space-around'}}>  
 <TouchableOpacity onPress={() =>**this**.\_onLogin()}>  
 <Text  
 style={{  
 color: '#ffffff',  
 fontSize: 20,  
 width: 100,  
 backgroundColor: 'green',  
 borderRadius: 2,  
 padding: 5  
 }}  
 >  
 login  
 </Text>  
 </TouchableOpacity>  
 <TouchableOpacity onPress={**this**.\_onRegister}>  
 <Text  
 style={{fontSize: 20, width: 100}}  
 >  
 register  
 </Text>  
 </TouchableOpacity>  
 </View>  
 </View>  
 );  
 }  
}

一个简单的登录页面，默认账号123密码 123 确定登录即可跳转。上面的用法之前都是学习到的。可参考之前的学习内容

在\_onlogin()方法中，点击登录按钮触发事件，满足登录条件，获取属性中的navigation进行跳转。跳转的目标页面随便创建就好了（我这边就不写出来了）。

接下来实现导航栈。

在入口js中：

**const** SimpleApp = StackNavigator({  
 Home: {screen: Login},  
 APP:{screen:App},  
});  
AppRegistry.registerComponent('ReactNativeApp', () => SimpleApp);

一共两个场景 Home 和APP。

效果：

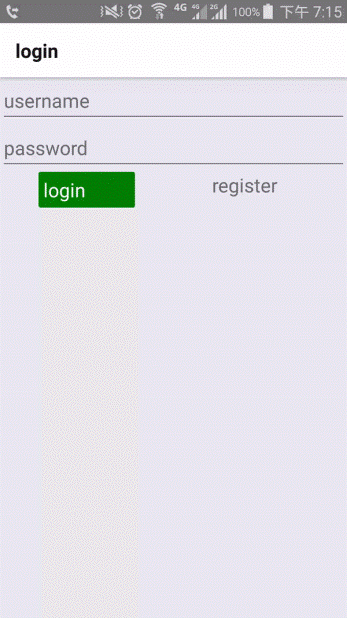


图1-1 效果