软著源代码

index.js:

```
/**
 * @format
 */

import {AppRegistry} from 'react-native';
import App from './App';
import {name as appName} from './app.json';

AppRegistry.registerComponent(appName, () => App);
```

App.js:

```
import * as React from 'react';
import { View, StyleSheet, Dimensions, FlatList, Image, TouchableOpacity, Text, ScrollView,
StatusBar } from "react-native";
import { NavigationContainer } from '@react-navigation/native';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
import { createStackNavigator } from '@react-navigation/stack'
import Ionicons from 'react-native-vector-icons/Ionicons';
import Global from './app/util/Global';
import TextStyles from './app/style/TextStyles';
import ViewStyles from './app/style/ViewStyles';
import ImageStyles, {gImageStyles} from './app/style/ImageStyles'
import HomePage from "./app/pages/HomePage";
import HotSpot from "./app/pages/HotSpot";
import Publish from "./app/pages/Publish";
import Convenience from "./app/pages/Convenience";
import My from "./app/pages/My";
import PersonInfo from "./app/pages/PersonInfo";
import Login from "./app/pages/Login"
import Message from "./app/pages/Message"
import Chat from "./app/pages/Chat"
import NewsDetail from "./app/pages/NewsDetail"
import WebPage from "./app/pages/WebPage";
import Search from "./app/pages/Search";
import { SafeAreaProvider } from 'react-native-safe-area-context';
import Splash from "./app/pages/Splash";
import Type from "./app/pages/Type";
import {WToast} from 'react-native-smart-tip'

const mainTab = createBottomTabNavigator();

function MainTab() {
    return(
        <mainTab.Navigator
            screenOptions={({ route }) => ({
                tabBarIcon: ({ focused, color, size }) => {
                    let iconName;
                    // 配置tabbaricon
                    if (route.name === '首页') {
                        return <Image source={require('./app/source/tap1.png')}
style={gImageStyles.tabbarItem}/>
                    } else if (route.name === '热点') {
                        return <Image source={require('./app/source/tap2.png')}
style={gImageStyles.tabbarItem}/>
                    } else if (route.name == '发布') {
                        return <Image source={require('./app/source/tap3.png')}
style={gImageStyles.middelTabbarItem}/>
                    } else if (route.name == '便民') {
                        return <Image source={require('./app/source/tap4.png')}
style={gImageStyles.tabbarItem}/>
                    } else if (route.name == '我的') {
                        return <Image source={require('./app/source/tap5.png')}
style={gImageStyles.tabbarItem}/>
                    }
                },
            })}

            tabBarOptions={{
                activeTintColor: '#FB5442',
                inactiveTintColor: '#D3D3D3',
            }}
        >
            <mainTab.Screen name="首页" component={HomePage} />
            <mainTab.Screen name="热点" component={HotSpot} />
            <mainTab.Screen name="发布" component={Publish} />
            <mainTab.Screen name="便民" component={Convenience} />
            <mainTab.Screen name="我的" component={My} />
        </mainTab.Navigator>
    );
}

const mainNav = createStackNavigator();

function MainNav() {
    return(
        <mainNav.Navigator
            headerMode={'none'}>
            <mainNav.Screen name='Login' component={Login}/>
            {/*tabbar部分*/}
            <mainNav.Screen name='MyTab' component={MainTab}/>
            {/*其他可跳转页面*/}
            <mainNav.Screen name='PersonInfo' component={PersonInfo}/>
            <mainNav.Screen name='Message' component={Message}/>
            <mainNav.Screen name='Chat' component={Chat}/>
            <mainNav.Screen name='NewsDetail' component={NewsDetail}/>
            <mainNav.Screen name='WebPage' component={WebPage}/>
            <mainNav.Screen name='Search' component={Search}/>
            <mainNav.Screen name='Type' component={Type}/>
        </mainNav.Navigator>
    );
}
```

```
export default function App() {
    const rootNav = createStackNavigator();
    return (
        <NavigationContainer>
            <rootNav.Navigator
                headerMode={'none'}
            >
                {/*启动广告*/}
                <rootNav.Screen name='Splash' component={Splash}/>
                <rootNav.Screen name="MainNav" component={MainNav}/>
            </rootNav.Navigator>
        </NavigationContainer>
    );
}
```

Login.js:

```
import React, { Component } from "react";
import { Text, View, StyleSheet, SafeAreaView, Image, TouchableOpacity } from "react-native";
import {gImageStyles} from "../style/ImageStyles";
import {gViewStyles} from "../style/ViewStyles";
import {gTextStyles} from "../style/TextStyles";

export default class Login extends Component {
    render() {
        return(
            <SafeAreaView style={gViewStyles.rootViewContainer}>
                <TouchableOpacity onPress={()=>{this.props.navigation.navigate('MyTab', null)}}>
                    <View style={gViewStyles.btnBack}>
                        <Image source={require('../source/guanbi.png')} style={gImageStyles.closeImg}/>
                    </View>
                </TouchableOpacity>
                <View style={styles.container}>
                    <Image source={require('../source/iconImg.jpg')} style={styles.iconImg}/>
                    <View style={styles.loginBtnContainer}>
                        <TouchableOpacity onPress={()=>{alert(1)}}>
                            <View style={gViewStyles.qqloginBack}>
                                <Image source={require('../source/qq-2.png')} style={gViewStyles.qqIcon}/>
                                <Text style={gTextStyles.loginText}>QQ登录</Text>
                            </View>
                        </TouchableOpacity>
                    </View>
                    <View style={[styles.loginBtnContainer, {marginBottom: 50}]}>
                        <TouchableOpacity onPress={()=>{alert(1)}}>
                            <View style={gViewStyles.wechatLoginBack}>
                                <Image source={require('../source/weixin.png')}
style={gViewStyles.wechatIcon}/>
                                <Text style={gTextStyles.loginText}>微信登录</Text>
                            </View>
                        </TouchableOpacity>
                    </View>
                </View>
            </SafeAreaView>
        );
    };
}
```

```
const styles = StyleSheet.create({
    container: {
        flex: 3,
        backgroundColor: 'white',
        alignItems: 'center',
    },
    iconImg: {
        flex: 2.5,
        width: gScreen.screen_width,
        resizeMode: 'center'
    },
    loginBtnContainer: {
        flex: 0.5,
        justifyContent: 'center',
    }

});
```

Splash.js:

```
import React, { Component } from "react";
import { Text, View, TouchableOpacity, StatusBar, Image, TouchableWithoutFeedback } from
"react-native";
import MyStatusBar from "../component/MyStatusBar";
import Swiper from 'react-native-swiper';
import {gViewStyles} from "../style/ViewStyles";
import {gImageStyles} from "../style/ImageStyles";
import {gTextStyles} from "../style/TextStyles";

export default class Splash extends Component {

    constructor(props) {
        super(props);
        this.state = {
            data: [require('../source/splashAd.jpg'), require('../source/ad6.jpg'), require('../source/
ad7.jpg')],
            load: true
        };
    }

    componentDidMount() {
        this.fetchADData();
        // setTimeout(this.onClickNext,3000);
    }

    onClickImg = (data) => {
        // if (item.href != null && item.href.length > 0) {
        //     let params = {
        //         url: item.href,
        //     };
        //     this.props.navigation.navigate('WebPage', params);
        // }
    }

    fetchADData = async() => {
        let params = {
            post_cate_id: 3,
        };
        let res = await httpApi.httpPost("http://dd.shenruxiang.com/api/v1/cate_adv_list", params);
        if (res.status == 0) {
            this.setState({
                data: res.data.data,
                load: true
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    render() {
        return(
            <View style={gViewStyles.rootViewContainer}>
                <StatusBar barStyle="dark-content"/>
                <Swiper
                    style={[gViewStyles.splashImgBack]}
                    horizontal={true}
                    activeDotColor={'#FB5442'}
                    paginationStyle={{bottom: 12}}
                    showsButtons={false}
                    autoplayTimeout={2}
                >

                    {
                        Array.isArray(this.state.data) ? (
                            Array.from(this.state.data).map((value, index, array) => {
                                return (
                                    <TouchableWithoutFeedback onPress={() => this.onClickImg(value)}
key={index}>
                                        {/*<Image source={{uri: value.src}} style={gImageStyles.splashImg}/>*/}
                                        <Image source={value} style={[{height: gScreen.screen_height, width:
gScreen.screen_width}]}/>
                                    </TouchableWithoutFeedback>
                                );
                            })
                        ) : (
                            <TouchableWithoutFeedback onPress={() => this.onClickImg(this.state.data)}>
                                {/*<Image source={{uri: this.state.data.src}} style={styles.img}/>*/}
                                <Image source={require('../source/splashAd.jpg')}
                                        style={gImageStyles.gImageStyles.splashImg}/>
                            </TouchableWithoutFeedback>
                        )
                    }
                </Swiper>
                <View style={gViewStyles.splashTextView}>
                    <Text style={gTextStyles.splashText}>跳过</Text>
                </View>
                <TouchableOpacity style={[gViewStyles.splashBtnView]} onPress={() => {
                    this.props.navigation.navigate('MainNav', null);
                }}>
                </TouchableOpacity>
            </View>
        );
    };
}
```

HomePage.js:

```
import React, { Component } from "react";
import { View, StyleSheet, Dimensions, FlatList, Image, TouchableOpacity, Text, ScrollView,
ActivityIndicator, RefreshControl } from "react-native";
import Ionicons from 'react-native-vector-icons/Ionicons';
import SearchView from "../component/SearchView";
import TitleListView from "../component/TitleListView";
import BannerView from "../component/BannerView"
import NewsView from "../component/news/NewsView"
import httpApi from "../tools/Api"
import LoadingView from "../component/LoadingView";
import ScrollableTabView, { ScrollableTabBar } from 'react-native-scrollable-tab-view';
import {gViewStyles} from "../style/ViewStyles";
import MyStatusBar from "../component/MyStatusBar";

export default class HomePage extends Component {

  // --------------------------- 生命周期 --------------------------------------
  constructor(props) {
    super(props);
    this.state = {
      newsTypeList: [],
      newsList: [],
      pageList: [],
      loaded: false,
      showIndex: 0,
      headerLoad: false,
      listLoad: true,
    };

  }

  async componentDidMount() {
    await this.fetchNewsTypeListData();

    // 默认请求第一个新闻列表
    this.showNewsList(this.state.newsTypeList[0]);
  }

  // ------------------------ 主题逻辑 -----------------------------

  // 点击头像
  onClickAvatar = (userId) => {
    let params = {
      userId: userId,
    };
    this.props.navigation.navigate('PersonInfo', params);
  };

  // 点击某个新闻分类标题
  onClickNewsType = (item) => {
    this.showNewsList(item);
  }

  // 点击广告
```

```
    onClickAd = (item) => {
        if (item.href != null && item.href.length > 0) {
            let params = {
                url: item.href,
            };
            this.props.navigation.navigate('WebPage', params);
        }
    }

    // 点击帖子
    onClickNews = (newsId) => {
        let params = {
            newsId: newsId,
        };
        this.props.navigation.navigate('NewsDetail', params);
    }

    // 展示新闻列表
    showNewsList = (item) => {
        if (this.state.newsList != null && this.state.newsList[item.id] != null) {

        } else {
            this.fetchNewsListData(item.id);
        }

        this.setState({
            showIndex: this.state.newsTypeList.findIndex(()=>item==item),
            headerLoad: false,
        });

    }

    // 跳转消息页
    onClickMesBtn = () => {
        this.props.navigation.navigate('Message');
    }

    // 点击tap
    onChangeTabs = (obj) => {
        let item = this.state.newsTypeList[obj.i];
        this.onClickNewsType(item);
    }

    // 搜索
    onSearch = (text) => {
        let params = {
            search: text
        };
        this.props.navigation.navigate('Search', params);
    }


    // ---------------------------- 网络请求 ------------------------------------------

    // 请求新闻分类列表
    fetchNewsTypeListData = async () => {
        let params = {};
        let res = await httpApi.getNewsTypeList(params);

        if (res.status == 0) {
            let newPageList = [];
            for (let i = 0; i < res.data.length; i++) {
                newPageList[res.data[i].id] = 1;
            }
            this.setState({
                pageList: newPageList,
                newsTypeList: res.data,
                loaded: true
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    // 下载具体新闻分类数据
    fetchNewsListData = async (newsTypeId) => {
        let params = {
            post_cate_id: newsTypeId,
        };
        let res = await httpApi.getNewsList(params);

        if (res.status == 0) {

            let newNewsList = this.state.newsList;
            newNewsList[newsTypeId] = res.data.data;

            this.setState({
                newsList: newNewsList,
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    // 上拉加载
    onLoadMoreData = async (value) => {

        if (this.state.listLoad == false) {
            return;
        }

        let params = '';
        params+=("post_cate_id=" + value.id + '&');
        params+=('page=' + (this.state.pageList[value.id]+=1) + '&');

        let res = await httpApi.httpPostWithParamsStr('http://dd.shenruxiang.com/api/v1/post_list',
params);
        if (res.status == 0) {

            if (res.data.data.length > 0) {
                let newNewsList = this.state.newsList;
                for (let i = 0; i < res.data.data.length; i++) {
                    newNewsList[value.id].push(res.data.data[i]);
        }
        this.setState({
          newsList: newNewsList,
        });
      } else {
        this.setState({
          listLoad: false,
        });
      }

    } else {
      alert("网络异常！请检查网络！");
    }
  }

// ------------------------- view 层 -----------------------------------
  render() {

    if (!this.state.loaded) {
      return <LoadingView/>;
    }

    return(
      <View style={gViewStyles.viewContainer1}>
        <MyStatusBar/>

        {/*搜索*/}
        <View style={styles.searchContainer}>
          <View style={styles.searchView}>
            <SearchView
              onSearch={this.onSearch}
            />
          </View>
          <TouchableOpacity onPress={this.onClickMesBtn}>
            <Image source={require('../source/xinxi.png')} style={styles.img3}/>
          </TouchableOpacity>
        </View>


        {/*新闻列表*/}
        <ScrollableTabView
          initialPage={this.state.showIndex}
          renderTabBar={() => <ScrollableTabBar style={styles.tabbar1} />}
          tabBarActiveTextColor='white'
          tabBarUnderlineStyle={styles.lineStyle}
          tabBarTextStyle={styles.text9}
          onChangeTab={this.onChangeTabs}
        >
          {this.state.newsTypeList.map((value, index, array) => {
            return (
              <View
                tabLabel={value.name}
              >
                <FlatList
                  data={this.state.newsList[value.id]}
                  renderItem={this.newsListItemView}
                  style={styles.newsList}
                  key={index}
```

```
                                    ListFooterComponent={()=>(
                                        this.state.listLoad == true ? (<View style={styles.indicatorContainer}>
                                            <ActivityIndicator
                                                style={styles.indicator}
                                                size={'small'}
                                                animating={true}
                                            />
                                        </View>) : (<View style={gViewStyles.bottomLoad}><Text>到底啦！ </
Text></View>)
                                        )}
                                    refreshControl={
                                        <RefreshControl
                                            refreshing={this.state.headerLoad}
                                            onRefresh={()=>{
                                                this.setState({
                                                    newsTypeList: [],
                                                    newsList: [],
                                                    pageList: [],
                                                    loaded: false,
                                                    showIndex: index,
                                                    listLoad: true,
                                                    headerLoad: true,
                                                });
                                                this.componentDidMount();
                                            }}
                                            enabled={true}
                                        />
                                    }
                                    onEndReached={()=>{
                                        this.onLoadMoreData(value, index);
                                    }}
                                    onEndReachedThreshold={0.1}
                                />
                            </View>


                        );
                    })}

                </ScrollableTabView>

            </View>
        );
    };


    newsListItemView = (item) => {
        if (item.item.is_adv) {
            return(
                <View>
                    <View style={styles.bottomSegmentation}></View>
                    <BannerView
                        onClickAd = {this.onClickAd}
                        data={item.item.data}
                    ></BannerView>
                </View>
                );
        } else {
            return (
            <NewsView
               onClickAvatar = {this.onClickAvatar}
               onClickNews={this.onClickNews}
               data={item.item.data}
               isShowPersonInfo={true}
            ></NewsView>
         );
      }

   };

}

const {width, height, scale} = Dimensions.get('window');

const styles = StyleSheet.create({
   container: {
      flex: 1,
      backgroundColor: 'white',
   },
   searchContainer: {
      flexDirection: 'row',
      alignItems: 'center',
      backgroundColor: '#FB5442'
   },
   searchView: {
      flex: gScreen.screen_width - 20,
   },
   segmentation: {
      marginTop: 3,
      height: 1,
      backgroundColor: gColor.grayLineColor
   },
   newsList: {

   },
   img3: {
      width: 30,
      height: 30,
      marginTop: 8,
      marginLeft: 8,
      marginRight: 10,
   },
   lineStyle: {
      height: 2,
      backgroundColor: 'white',
   },
   tabbar1: {
      backgroundColor: '#FB5442',
   },
   text9: {
      color: 'white',
      fontSize: 16
   },
   indicatorContainer:{
      alignItems:'center'
   },
   indicator:{
      color:'red',
      margin:10
```

```
    },
    bottomSegmentation: {
        height: 10,
        backgroundColor: gColor.grayLineColor
    },
});
```

HotSpot.js:

```
import React, { Component } from "react";
import {Text, View, FlatList, StyleSheet, Dimensions, StatusBar} from "react-native";
import BannerView from "../component/BannerView"
import NewsView from "../component/news/NewsView"
import httpApi from "../tools/Api";
import LoadingView from "../component/LoadingView";
import MyNavigationBar from '../component/MyNavigationBar'
import MyStatusBar from "../component/MyStatusBar";
import {gViewStyles} from "../style/ViewStyles";

export default class HotSpot extends Component {

    // --------------------------- 生命周期 -------------------------------------
    constructor(props) {
        super(props);
        this.state = {
            newsList: [],
            load: false
        };
    }

    componentDidMount() {
        this.fetchNewsListData()
    }

    // 点击帖子
    onClickNews = (newsId) => {
        let params = {
            newsId: newsId,
        };
        this.props.navigation.navigate('NewsDetail', params);
    }

    // 点击头像
    onClickAvatar = (userId) => {
        let params = {
            userId: userId,
        };
        this.props.navigation.navigate('PersonInfo', params);
    };

    // 点击广告
    onClickAd = (item) => {
        if (item.href != null && item.href.length > 0) {
            let params = {
                url: item.href,
            };
            this.props.navigation.navigate('WebPage', params);
        }
    }

    // 下载具体新闻分类数据
    fetchNewsListData = async () => {
        let params = {
            post_cate_id: 2,
        };
        let res = await httpApi.getNewsList(params);

        if (res.status == 0) {

            this.setState({
                newsList: res.data.data,
                load: true
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    render() {

        if (!this.state.load) {
            return <LoadingView/>;
        }

        return(
            <View style={gViewStyles.rootViewContainer}>
                <MyStatusBar/>
                <MyNavigationBar
                    title={'热点'}
                    hiddenBack={true}
                ></MyNavigationBar>
                {/*新闻列表*/}
                <FlatList
                    data={this.state.newsList}
                    renderItem={this.newsListItemView}
                    style={styles.newsList}
                />
            </View>
        );
    };

    newsListItemView = (item) => {
        if (item.item.is_adv) {
            return(<BannerView
                onClickAd = {this.onClickAd}
                data={item.item.data}
            ></BannerView>);
        } else {
            return (
                <NewsView
                    onClickAvatar = {this.onClickAvatar}
                    data={item.item.data}
                    onClickNews={this.onClickNews}
                    isShowPersonInfo={true}
            ></NewsView>
        );
    }

};

}

const {width, height, scale} = Dimensions.get('window');

const styles = StyleSheet.create({
    container: {
        flex: 1,
        backgroundColor: 'white'
    },
    newsList: {
        backgroundColor: 'white'
    }
});


Publish.js:
import React, {Component} from "react";
import {Text, View, StyleSheet, TextInput, Image, Dimensions, ScrollView, TouchableOpacity,
FlatList, Alert, DeviceEventEmitter, ActivityIndicator} from "react-native";
import ImagePicker from 'react-native-image-picker';
import MyNavigationBar from '../component/MyNavigationBar'
import MyStatusBar from "../component/MyStatusBar";
import {gViewStyles} from "../style/ViewStyles";
import httpApi from "../tools/Api";


export default class Publish extends Component {

    constructor(props) {
        super(props);
        this.state = {
            imageArr: [], // 动态图片
            post_content: null, // 动态内容
            post_cate_id: null, // 分类id
            phone_model: null, // 手机型号
            post_position: null, // 地点
            post_type: 1, // 帖子类型 1图片 2视频
            typeData: null, // 分类数据
            load: false
        };
        this.emitter = null;
    }

    componentDidMount(){
        this.emitter = DeviceEventEmitter.addListener('updateType', (type)=>{this.setState({
            post_cate_id: type.id, // 分类id
            typeData: type, // 分类数据
        })});
    };
    //在组件销毁的时候要将其移除
    componentWillUnmount(){
        this.emitter.remove();
    };

    // 发布帖子
    onPublish = async () => {

        this.setState({
            load: true
        });

        let imageArrStr = '';
        for (let i = 0; i < this.state.imageArr.length; i++) {
            let subImgStr = 'image' + i + '=' + this.state.imageArr[i].uri + '&';
            imageArrStr+=subImgStr;
        }

        let params = '';
        params+=imageArrStr;
        params+=("post_content=" + this.state.post_content + '&');
        params+=('post_cate_id=' + this.state.post_cate_id + '&');
        params+=('phone_model=' + this.state.phone_model + '&');
        params+=('post_position=' + this.state.post_position + '&');
        params+=('post_type=' + this.state.post_type + '&');

        let res = await httpApi.httpPostWithParamsStr("http://dd.shenruxiang.com/api/v1/
post_send", params);
        if (res.status == 0) {
            this.setState({
                load: false
            });
            WToast.show({data: '发布成功'});
        } else {
            alert("网络异常！请检查网络！");
        }

    }

    // 删除照片
    deleteImg = (index) => {

        Alert.alert(
            '提示', //提示标题
            '是否删除该照片', //提示内容
            [
                { text: '取消', onPress: () => console.log('点击取消') },
                {
                    text: '确定', onPress: () => {
                        let newImgArr = this.state.imageArr;
                        newImgArr.splice(index, 1);
                        this.setState({
                            imageArr: newImgArr,
                        });
                    }
                }
            ] //按钮集合
        )

    }

    // open album
    onSelectImg = () => {

        const options = {
            title: '请选择图片来源',
            cancelButtonTitle:'取消',
            takePhotoButtonTitle:'拍照',
            chooseFromLibraryButtonTitle:'相册图片',
            mediaType: 'photo',
            maxWidth: 700,
            maxHeight: 700,
            storageOptions: {
                skipBackup: true,
                path: 'images',
            },
        };

        ImagePicker.showImagePicker(options, (response) => {

            if (response.didCancel) {
            } else if (response.error) {
            } else {
                const source = {uri: 'data:image/jpeg;base64,'+ response.data};
                let newImgArr = this.state.imageArr;
                newImgArr.push(source);
                this.setState({
                    imageArr: newImgArr,
                });
            }
        });
    }

    // open type page
    onClickType = () => {
        this.props.navigation.navigate('Type');
    }

    render() {
        return (
            <View style={gViewStyles.rootViewContainer}>
                <MyStatusBar/>
                <MyNavigationBar
                    title={'动态发布'}
                    hiddenBack={true}
                ></MyNavigationBar>
                <ScrollView style={styles.scorllviewContainer}>
                    <View style={styles.topContainer}>
                        <TextInput
                            placeholder={'输入你想说的话。。。'}
                            placeholderTextColor={gColor.grayTextColor}
                            underlineColorAndroid={'transparent'}
                            style={styles.textInput}
                            underlineColorAndroid="transparent"
                            multiline={true}
                            onChangeText={(text) => this.setState({post_content:text})}
                        />

                        <FlatList
                            extraData={this.state}
                            ListHeaderComponent={this.selectImgView}
                            data={this.state.imageArr}
                            renderItem={({item, index})=>(<TouchableOpacity
            onPress={()=>this.deleteImg(index)}>
                                <Image source={item} style={styles.selectImg} />
                            </TouchableOpacity>)}
                            horizontal={true}
                        >
                        </FlatList>
                    </View>


                    <View style={gViewStyles.line2}></View>

                    {/*选择分类*/}
                    <TouchableOpacity onPress={()=>this.onClickType()}>
                        <View style={styles.subContainer}>
                            <Image source={require('../source/fenlei.png')} style={styles.itemIcon}/>
                            <Text style={styles.itemText}>选择分类</Text>
                            {this.state.typeData != null ? (<Text style={styles.itemText1}
            >{this.state.typeData.name}</Text>) : (<Text style={styles.itemText1}>默认发布到最新板块</
            Text>)}
                            <Image source={require('../source/youjiantou.png')} style={styles.itemArrow}/>
                        </View>
                    </TouchableOpacity>


                    <View style={styles.line1}></View>

                    {/*选择定位*/}
                    <View style={styles.subContainer}>
                        <Image source={require('../source/dingwei.png')} style={styles.itemIcon}/>
                        <Text style={styles.itemText}>选择定位</Text>
                        <Text style={styles.itemText1}>确认定位已开启</Text>
                        <Image source={require('../source/youjiantou.png')} style={styles.itemArrow}/>
                    </View>

                    <View style={styles.line1}></View>

                    {/*确认发布*/}
                    <View style={styles.subContainer1}>
                        <TouchableOpacity onPress={()=>this.onPublish()}>
                            <View style={styles.backView}>
                                <Text style={styles.text7}>确认发布</Text>
                            </View>
                        </TouchableOpacity>
                    </View>
                </ScrollView>
```

```
                {this.state.load == true ? (<ActivityIndicator size="large" color='#FB5442'
style={gViewStyles.loading}/>
                ) : (<View></View>)}
            </View>
        );
    };

    selectImgView = ({}) => {
        return(
            <TouchableOpacity onPress={() => this.onSelectImg()}>
                <Image source={require('../source/camera.jpg')} style={styles.selectImg}/>
            </TouchableOpacity>
        );
    }

}

const {width, height, scale} = Dimensions.get('window');

const styles = StyleSheet.create({
    scorllviewContainer: {
        backgroundColor: '#F4F1F4',
        flex: 1,
    },
    textInput: {
        textAlignVertical: 'top',
        padding: 5,
        margin: 5,
        height: 140,
        fontSize: 14,
    },
    topContainer: {
        marginTop: 10,
        flex: 1,
        height: 250,
        backgroundColor: 'white',
    },
    selectImg: {
        margin: 5,
        height: 60,
        width: 60,
    },
    line1: {
        marginLeft: 10,
        marginRight: 10,
        height: 1,
        backgroundColor: gColor.grayLineColor
    },
    subContainer: {
        height: 50,
        flexDirection: 'row',
        alignItems: 'center',
        backgroundColor: 'white'
    },
    itemIcon: {
        flex: 50,
        width: 18,
        marginLeft: 10,
    },
    itemText: {
        flex: 200,
        marginLeft: 10,
    },
    itemText1: {
        flex: 200,
        marginLeft: 20,
        color: gColor.grayTextColor
    },
    subContainer1: {
        marginTop: 30,
        height: 40,
        alignItems: 'center',
        justifyContent: 'center',
        width: gScreen.screen_width,
        marginBottom: 40,
    },
    confirmPublish: {
        width: 300,
        height: 40,
    },
    backView: {
        alignItems: 'center',
        justifyContent: 'center',
        width: 200,
        height: 40,
        borderRadius: 5,
        backgroundColor: gColor.orangeBackColor,
    },
    text7: {
        color: 'white',
        fontSize: 14,
    }
});
```

Convenience.js:

```
import React, { Component } from "react";
import { Image, View, SafeAreaView, FlatList, TouchableWithoutFeedback } from "react-native";
import MyNavigationBar from '../component/MyNavigationBar'
import MyStatusBar from "../component/MyStatusBar";
import {gImageStyles} from "../style/ImageStyles";
import {gViewStyles} from "../style/ViewStyles";

export default class Convenience extends Component {
    render() {
        return(
            <View style={gViewStyles.rootViewContainer}>
                <MyStatusBar/>
                <SafeAreaView style={[gViewStyles.rootViewContainer, backgroundColor='#F4F1F4']}>
                    <MyNavigationBar
                        title={'便民'}
                        hiddenBack={true}
                    ></MyNavigationBar>
                    <FlatList
                        style={backgroundColor='#F4F1F4'}
                        data={[require('../source/ad1.jpg'),
                            require('../source/ad2.jpg'),
                            require('../source/ad3.jpg'),
                            require('../source/ad4.jpg'),
                            require('../source/ad5.jpg'),
                            require('../source/ad6.jpg'),
                            require('../source/ad7.jpg'),
                            require('../source/splashAd.jpg'),
                            require('../source/splashAd1.png'),]}
                        renderItem={({item})=>{
                            let imgW = Image.resolveAssetSource(item).width;
                            let imgH = Image.resolveAssetSource(item).height;
                            let showW = gScreen.screen_width - 24;
                            let showH = showW / imgW * imgH;
                                return(<View style={gViewStyles.itemView1}>
                                    <TouchableWithoutFeedback>
                                        <View>
                                            <Image source={item}
                                                    style={[gImageStyles.ad1, {height: showH}]}></Image>
                                            <View style={gViewStyles.segmentation}></View>
                                        </View>
                                    </TouchableWithoutFeedback>
                                </View>);
                        }}
                    >
                    </FlatList>
                </SafeAreaView>
            </View>
        );
    };
}
```

```
My.js:
import React, { Component } from "react";
import {Text, View, StyleSheet, Image, ImageBackground, FlatList, Dimensions, TouchableOpacity
} from "react-native";
import httpApi from "../tools/Api";
import LoadingView from "../component/LoadingView";

export default class My extends Component {

    constructor(props) {
        super(props);
        this.state = {
            optionData: [
                {title: '我的动态', img: require('../source/dongtai.png')},
                {title: '我的相册', img: require('../source/xiangce.png')},
                {title: '实名认证', img: require('../source/shiming.png')},
                {title: '广告投放', img: require('../source/guanggao.png')},
                {title: '帮助中心', img: require('../source/bangzhu.png')},
                {title: '投诉反馈', img: require('../source/tousu.png')},
                {title: '在线客服', img: require('../source/kefu.png')},
                {title: '关于我们', img: require('../source/guanyu.png')},
                {title: '检查更新', img: require('../source/gengxin.png')},
                {title: '退出登录', img: require('../source/tuichu.png')},
            ],
            personalData: [],
            load: false,
        };
    }

    componentDidMount() {
        this.fetchPersonalData();
    }

    // 请求个人数据
    fetchPersonalData = async () => {
        let params = {};
        let res = await httpApi.getPersonalData(params);

        if (res.status == 0) {
            this.setState({
                personalData: res.data,
                load: true
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    render() {
        if (this.state.load == false) {
            return <LoadingView></LoadingView>
        }
        return(
            <View style={styles.container}>
                {this.topView()}
                <FlatList
                    style={styles.list}
                    renderItem={({item, index})=>(<TouchableOpacity onPress={()=>{
                        if (item.title == '我的动态') {
                            let params = {
                                userId: 2,
                            };
                            this.props.navigation.navigate('PersonInfo', params);
                        } else if (item.title == '我的相册') {
                            let params = {
                                userId: 2,
                            };
                            this.props.navigation.navigate('PersonInfo', params);
                        } else if (item.title == '实名认证') {

                        } else if (item.title == '广告投放') {

                        } else if (item.title == '帮助中心') {

                        } else if (item.title == '投诉反馈') {

                        } else if (item.title == '在线客服') {

                        } else if (item.title == '关于我们') {
                        } else if (item.title == '检查更新') {

                        } else {

                        }
                    }}>
                        <View style={styles.itemContainer}>
                            <View style={styles.itemInfoContainer}>
                                <Image source={item.img}
                                        style={styles.itemIcon}/>
                                <Text style={styles.itemText}>{item.title}</Text>
                                <Image source={require('../source/youjiantou.png')} style={styles.itemArrow}/>
                            </View>
                            <View style={styles.bottomLine}></View>
                        </View>
                    </TouchableOpacity>)}
                    data={this.state.optionData}
                />
            </View>
        );
    };

    topView = () => {
        return(
            <ImageBackground style={styles.topContainer} source={require('../source/
wodebeijingtu.png')}>
                <Image source={{uri: this.state.personalData.user_info.avatar}} style={styles.avatar}/>
                <View style={styles.personalInfoContainer}>
                    <Text style={styles.nickName}>{this.state.personalData.user_info.nick_name}</Text>
                    <Text style={styles.commonText}>ID：{this.state.personalData.user_info.id}</Text>
                    <Text style={styles.commonText}>粉丝：{this.state.personalData.user_info.fan_num}
关注：{this.state.personalData.user_info.follow_num}</Text>
                </View>
                <Image source={require('../source/xiugai.png')} style={styles.updateNickName}/>
            </ImageBackground>
        );
    }

}

const {width, height, scale} = Dimensions.get('window');

const styles = StyleSheet.create({
    container: {
        backgroundColor: 'white',
        flex: 1,
    },
    topContainer: {
        height: 150,
        flexDirection: 'row',
        paddingTop: gScreen.statusBarHeight,
    },
    avatar: {
        height: 60,
        width: 60,
        margin: 16,
        borderRadius: 30,
    },
    personalInfoContainer: {
        marginTop: 15,
    },
    nickName: {
        fontSize: 18,
        color: 'white'
    },
    commonText: {
        color: 'white'
    },
    updateNickName: {
        margin: 10,
    },
    sectionHeader: {
        paddingTop: 2,
        paddingLeft: 10,
        paddingRight: 10,
        paddingBottom: 2,
    },
    itemContainer: {
        padding: 10,
        fontSize: 18,
        height: 48,
    },
    itemIcon: {
        flex: 20,
        marginTop: 10,
        width: 16,
    },
    bottomLine: {
        marginTop: 13,
        height: 1,
        backgroundColor: gColor.grayLineColor
    },
    itemInfoContainer: {
        flexDirection: 'row',
        alignItems: 'center'
    },
    itemText: {
        flex: width - 200,
        marginTop: 11,
        marginLeft: 10,
    },
    itemArrow: {
        height: 25,
        width: 25,
        marginTop: 10,
    },
    list: {
        backgroundColor: 'white',
    }
});
```

Chat.js:

```
import React, { Component } from "react";
import {
    Text,
    View,
    StyleSheet,
    SafeAreaView,
    Image,
    TouchableOpacity,
    Dimensions,
    TextInput,
    FlatList,
    Keyboard
} from "react-native";
import MyNavigationBar from '../component/MyNavigationBar'
import MyStatusBar from "../component/MyStatusBar";
import {gViewStyles} from "../style/ViewStyles";
import {gImageStyles} from "../style/ImageStyles";
import {gTextStyles} from "../style/TextStyles";
import httpApi from "../tools/Api";
import LoadingView from "../component/LoadingView";

export default class Chat extends Component {

    constructor(props) {
        super(props);
        this.state = {
            data: null,
            load: false,
            keyboardHeight: 0,
        };

        this.keyboardWillShowListener = null;
        this.keyboardWillHideListener = null;
    }

    componentDidMount() {
        this.fetchChatData();
        this.keyboardWillShowListener =
Keyboard.addListener('keyboardWillShow',this.keyboardDidShow);
        this.keyboardWillHideListener =
Keyboard.addListener('keyboardWillHide',this.keyboardDidHide);
    }


    //注销监听
    componentWillUnmount () {
        this.keyboardWillShowListener && this.keyboardWillShowListener.remove();
        this.keyboardWillHideListener && this.keyboardWillHideListener.remove();
    }


    //键盘弹起后执行
    keyboardDidShow = (event) =>  {
        this.setState({
            keyboardHeight: event.endCoordinates.height,
        });
    }

    //键盘收起后执行
    keyboardDidHide = () => {
        this.setState({
            keyboardHeight: 0,
        });
    }

    onEndEditing = async (text) => {
        let params = '';
        params+=("to_user_id=" + this.props.route.params.userId + '&');
        params+=('content=' + text + '&');
        params+=('content_type=' + 1 + '&');

        let res = await httpApi.httpPostWithParamsStr('http://dd.shenruxiang.com/api/v1/
user_send_message', params);
        if (res.status == 0) {
            this.fetchChatData();
            this.refs.textInput.clear();
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    fetchChatData = async () => {
        let params = '';
        params+=("to_user_id=" + this.props.route.params.userId + '&');

        let res = await httpApi.httpPostWithParamsStr('http://dd.shenruxiang.com/api/v1/
user_message_history_list', params);
        if (res.status == 0) {
            this.setState({
                data: res.data,
                load: true,
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    render() {
        if (!this.state.load) {
            return <LoadingView/>;
        }
        return(

            <View style={gViewStyles.rootViewContainer}>
                <MyStatusBar></MyStatusBar>
                <SafeAreaView style={gViewStyles.grayRootViewContainer1}>
                    <MyNavigationBar
                        title={'私信'}
                        onClickBack={()=>{this.props.navigation.goBack();}}
                    ></MyNavigationBar>
                    <View style={styles.chat}>
                        <FlatList
                            ref = "flatList"
                            onContentSizeChange={()=> this.refs.flatList.scrollToEnd()}
                            data={this.state.data}
                            renderItem={this.chatItemView}
                            style={styles.chatList}
                        ></FlatList>
                    </View>
                    <View style={[styles.input, {marginBottom: this.state.keyboardHeight}]}>
                        <TextInput underlineColorAndroid="transparent" placeholder="这里输入消息"
                                    ref='textInput'
                                    style={styles.searchTextInput}
                                    onSubmitEditing={(event)=>this.onEndEditing(event.nativeEvent.text)}>
                        </TextInput>
                        {/*<View style={styles.input2}>*/}
                        {/*<TouchableOpacity onPress={()=>{alert(1)}}>*/}
                        {/*<Image source={require('../source/attention1.jpg')} style={styles.inputIcon}/>*/}
                        {/*</TouchableOpacity>*/}
                        {/*<TouchableOpacity onPress={()=>{alert(1)}}>*/}
                        {/*<Image source={require('../source/attention1.jpg')} style={styles.inputIcon}/>*/}
                        {/*</TouchableOpacity>*/}
                        {/*<TouchableOpacity onPress={()=>{alert(1)}}>*/}
                        {/*<Image source={require('../source/attention1.jpg')} style={styles.inputIcon}/>*/}
                        {/*</TouchableOpacity>*/}
                        {/*<TouchableOpacity onPress={()=>{alert(1)}}>*/}
                        {/*<Image source={require('../source/attention1.jpg')} style={styles.inputIcon}/>*/}
                        {/*</TouchableOpacity>*/}
                        {/*<TouchableOpacity onPress={()=>{alert(1)}}>*/}
                        {/*<Image source={require('../source/attention1.jpg')} style={styles.inputIcon}/>*/}
                        {/*</TouchableOpacity>*/}
                        {/*</View>*/}
                    </View>
                </SafeAreaView>
            </View>
        );
    };

    chatItemView = ({item}) => {
        if (item.user_id == 2) {
            return(this.rightChatItemView({item}));
        } else {
            return(this.leftChatItemView({item}));
        }
    }

    leftChatItemView = ({ item }) => {
        return(
            <View style={styles.chatItemView1}>
                <TouchableOpacity>
                    <Image source={{uri: item.user.avatar}} style={styles.avatar}/>
                </TouchableOpacity>
                <View style={styles.leftMes}>
                    <Text style={styles.mes1}>{item.content}</Text>
                </View>

            </View>
        );
    }

    rightChatItemView = ({ item }) =>  {
        return(<View style={styles.chatItemView2}>
            <View style={styles.rightMes}>
                <Text style={styles.mes2}>{item.content}</Text>
            </View>
            <TouchableOpacity>
                <Image source={{uri: item.user.avatar}} style={styles.avatar}/>
            </TouchableOpacity>
        </View>);
    }
}

const {width, height, scale} = Dimensions.get('window');

const styles = StyleSheet.create({
    container: {
        flex: 1,
    },
    chat: {
        flex: height - 70
    },
    input: {
        flex: 70,
    },
    searchTextInput: {
        padding: 5,
        height: 35,
        margin: 10,
        backgroundColor: '#fff',

        borderRadius: 4,
    },
    input2: {
        height: 30,
        flexDirection: 'row',
        justifyContent: 'space-around'
    },
    inputIcon: {
        height: 30,
        width: 30,
    },
    chatList: {
        margin: 5,
    },
    chatItemView1: {
        flexDirection: 'row',
    },
    chatItemView2: {
        flexDirection: 'row',
        justifyContent: 'flex-end',
    },
    avatar: {
        height: 50,
        width: 50,
        margin: 5,
        borderRadius: 25,
    },
    leftMes: {
        marginTop: 10,
        width: width - 200,
        backgroundColor: 'white',
        borderRadius: 10,
    },
    rightMes: {
        marginTop: 10,
        width: width - 200,
        backgroundColor: '#FB5442',
        borderRadius: 10,
    },
    mes1: {
        padding: 10,
        fontSize: 14,
        color: '#333333',
    },
    mes2: {
        padding: 10,
        color: 'white',
        fontSize: 14,
    }
});
```

Message.js:

```
import React, { Component } from "react";
import {Text, View, StyleSheet, SafeAreaView, Image, FlatList, TouchableOpacity, Dimensions}
from "react-native";
import httpApi from "../tools/Api";
import LoadingView from "../component/LoadingView";
import MyNavigationBar from '../component/MyNavigationBar'
import MyStatusBar from "../component/MyStatusBar";
import ScrollableTabView, {DefaultTabBar,ScrollableTabBar} from 'react-native-scrollable-tab-
view';
import {gViewStyles} from "../style/ViewStyles";
import {gTextStyles} from "../style/TextStyles";

export default class Message extends Component {

    constructor(props) {
        super(props);
        this.state = {
            data: [[], [], [], [], []],
            titleList: ['关注', '评论', '点赞', '私信', '通知'],
            load: false,
            showIndex: 0
        };
    }

    componentDidMount() {
        this.fetchFansData();
    }

    // 点击头像
    onClickAvatar = (item) => {
        let params = {
            userId: item.to_user.id,
        };
        this.props.navigation.navigate('PersonInfo', params);
    };

    // 点击关注
    onClickAttention = (item) => {
        // 跳转个人详情
        let params = {
            userId: item.user.id,
        };
        this.props.navigation.navigate('PersonInfo', params);
    }

    // 点击评论
    onClickComment = (item) => {
        // 跳转帖子详情
        let params = {
            userId: item.post_id,
        };
        this.props.navigation.navigate('NewsDetail', params);
    }

    // 点击关注
    onClickPraise = (item) => {
        // 跳转帖子详情
        let params = {
            userId: item.post_id,
        };
        this.props.navigation.navigate('NewsDetail', params);
    }

    // 请求粉丝列表
    fetchFansData = async() => {
        let params = {};
        let res = await httpApi.httpPost("http://dd.shenruxiang.com/api/v1/user_fan_list", params);
        if (res.status == 0) {
            let newData = this.state.data;
            newData[0] = res.data;
            this.setState({
                data: newData,
                showIndex: 0,
                load: true
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    // 请求评论列表
    fetchCommentData = async() => {
        let params = {};
        let res = await httpApi.httpPost('http://dd.shenruxiang.com/api/v1/user_comment_list',
params);
        this.updateData(1, res);
    }

    // 请求关注列表
    fetchAttentionData = async() => {
        let params = {};
        let res = await httpApi.httpPost('http://dd.shenruxiang.com/api/v1/user_follow_list', params);
        if (res.status == 0) {
            let newData = this.state.data;
            newData[2] = res.data;
            this.setState({
                data: newData,
        showIndex: 2,
        load: true
      });
    } else {
      alert("网络异常！请检查网络！");
    }
  }

  // 请求私信列表
  fetchChatData = async() => {
    let params = {};
    let res = await httpApi.httpPost('http://dd.shenruxiang.com/api/v1/user_message_list',
params);
    if (res.status == 0) {
      let newData = this.state.data;
      newData[3] = res.data;
      this.setState({
        data: newData,
        showIndex: 3,
        load: true
      });
    } else {
      alert("网络异常！请检查网络！");
    }
  }

  // 请求通知列表
  fetchNoticeData = async () => {
    let params = {};
    let res = await httpApi.httpPost('http://dd.shenruxiang.com/api/v1/notice_list', params);
    this.updateData(4, res);
  }

  updateData = (index, res) => {
    if (res.status == 0) {
      let newData = this.state.data;
      newData[index] = res.data.data;
      this.setState({
        data: newData,
        showIndex: index,
        load: true
      });
    } else {
      alert("网络异常！请检查网络！");
    }
  }

  // 切换tab
  onChangeTabs = (index) => {

    if (this.state.data != null && this.state.data[index].length > 0) {
      // 有数据
      let newData = this.state.data;
      this.setState({
        data: newData,
        showIndex: index,
                load: true
            });
        } else {
            // 无数据
            this.fetchData(index);
        }

    }

    fetchData = (index) => {
        if (index == 0) {
            this.fetchFansData();
        } else if (index == 1) {
            this.fetchCommentData();
        } else if (index == 2) {
            this.fetchAttentionData();
        } else if (index == 3) {
            this.fetchChatData();
        } else if (index == 4) {
            this.fetchNoticeData();
        }
    }

    onClickChat = (item) => {
        let params = {
            userId: item.to_user_id,
        };
        this.props.navigation.navigate('Chat', params);
    }

    render() {
        if (this.state.load == false) {
            return <LoadingView></LoadingView>
        }
        return (
            <View style={gViewStyles.rootViewContainer}>
                <MyStatusBar></MyStatusBar>
                <SafeAreaView style={styles.container}>
                    <MyNavigationBar
                        title={'消息记录'}
                        onClickBack={()=>{this.props.navigation.goBack();}}
                    ></MyNavigationBar>

                    {this.headerItemView()}

                    {this.listView()}

                </SafeAreaView>
            </View>

        );
    };

    listView = () => {
        return(
            <FlatList
                extraData={this.state}
                data={this.state.data[this.state.showIndex]}
                renderItem={this.listItemView}
                style={gViewStyles.flatList}
            />
        );
    }

    headerItemView = () => {
        return(
            <View style={styles.topContainer}>
                <View style={styles.msgIconContainer}>
                    <TouchableOpacity onPress={()=>this.onChangeTabs(0)}>
                        <Image source={require('../source/guanzhu.png')} style={styles.msgIcon}/>
                    </TouchableOpacity>
                    <Text style={styles.msgDes}>粉丝</Text>
                    {this.state.showIndex == 0 ? (<View style={gViewStyles.scrollBarBottomLine1}></
View>) : (<View></View>)}
                </View>

                <View style={styles.msgIconContainer}>
                    <TouchableOpacity onPress={()=>this.onChangeTabs(1)}>
                        <Image source={require('../source/pinglun.png')} style={styles.msgIcon}/>
                    </TouchableOpacity>
                    <Text style={styles.msgDes}>评论</Text>
                    {this.state.showIndex == 1 ? (<View style={gViewStyles.scrollBarBottomLine1}></
View>) : (<View></View>)}
                </View>

                <View style={styles.msgIconContainer}>
                    <TouchableOpacity onPress={()=>this.onChangeTabs(2)}>
                        <Image source={require('../source/dianzan.png')} style={styles.msgIcon}/>
                    </TouchableOpacity>
                    <Text style={styles.msgDes}>关注</Text>
                    {this.state.showIndex == 2 ? (<View style={gViewStyles.scrollBarBottomLine1}></
View>) : (<View></View>)}
                </View>

                <View style={styles.msgIconContainer}>
                    <TouchableOpacity onPress={()=>this.onChangeTabs(3)}>
                        <Image source={require('../source/pinglun.png')} style={styles.msgIcon}/>
                    </TouchableOpacity>
                    <Text style={styles.msgDes}>私信</Text>
                    {this.state.showIndex == 3 ? (<View style={gViewStyles.scrollBarBottomLine1}></
View>) : (<View></View>)}
                </View>

                <View style={styles.msgIconContainer}>
                    <TouchableOpacity onPress={()=>this.onChangeTabs(4)}>
                        <Image source={require('../source/tongzhi.png')} style={styles.msgIcon}/>
                    </TouchableOpacity>
                    <Text style={styles.msgDes}>通知</Text>
                    {this.state.showIndex == 4 ? (<View style={gViewStyles.scrollBarBottomLine1}></
View>) : (<View></View>)}
                </View>
            </View>
        );
    }
    listItemView = ({item}) => {
        switch (this.state.showIndex) {
            case 0:
            {
                return(this.fansItemView(item));
            }
                break;
            case 1:
            {
                return(this.commentItemView(item));
            }
                break;
            case 2:
            {
                return(this.attentionItemView(item));
            }
                break;
            case 3:
            {
                return(this.chatItemView(item));
            }
                break;
            case 4:
            {
                return(this.noticeItemView(item));
            }
                break;
            default:
            {
                return(<View></View>);
            }
                break;
        }
    }

    attentionItemView = (item) => {
        return(
            <View>
                <TouchableOpacity onPress={()=>this.onClickAttention(item)}>
                    <View style={styles.line}></View>
                    <View style={styles.msgContainer}>
                        {/*<TouchableOpacity onPress={()=>this.onClickAvatar(item)}>*/}
                            <Image source={{uri: item.to_user.avatar}} style={styles.avatar}/>
                            {/*<Image source={require('../source/未登陆.png')} style={styles.avatar}/>*/}
                        {/*</TouchableOpacity>*/}
                        <View style={styles.msgSubContainer}>
                            <View style={styles.mesSubContainer}>
                                <Text style={styles.text1}>{item.to_user.nick_name}</Text>
                            </View>
                            <View style={styles.mesSubContainer1}>
                                <Text style={styles.text2}>你关注了Ta！</Text>
                            </View>
                        </View>
                    </View>
                </TouchableOpacity>
            </View>
        );
    }
```

```
fansItemView = (item) => {
  return(
    <View>
      <TouchableOpacity onPress={()=>this.onClickAttention(item)}>
        <View style={styles.line}></View>
        <View style={styles.msgContainer}>
          {/*<TouchableOpacity onPress={()=>this.onClickAvatar(item)}>*/}
          <Image source={{uri: item.user.avatar}} style={styles.avatar}/>
          {/*<Image source={require('../source/未登陆.png')} style={styles.avatar}/>*/}
          {/*</TouchableOpacity>*/}
          <View style={styles.msgSubContainer}>
            <View style={styles.mesSubContainer}>
              <Text style={styles.text1}>{item.user.nick_name}</Text>
            </View>
            <View style={styles.mesSubContainer1}>
              <Text style={styles.text2}>Ta关注了你！ </Text>
            </View>
          </View>
        </View>
      </TouchableOpacity>
    </View>
  );
}

commentItemView = (item) => {
  return(
    <View>
      <TouchableOpacity onPress={()=>this.onClickComment(item)}>
        <View style={styles.line}></View>
        <View style={styles.msgContainer}>
          <TouchableOpacity>
            {/*<Image source={require('../source/未登陆.png')} style={styles.avatar}/>*/}
            <Image source={{uri: item.user.avatar}} style={styles.avatar}/>
          </TouchableOpacity>
          <View style={styles.msgSubContainer}>
            <View style={styles.mesSubContainer}>
              <Text style={styles.text1}>{item.user.nick_name}</Text>
              <View style={styles.container1}>
                <Text style={styles.text3}>1小时前</Text>
                <Text style={styles.text2}> * </Text>
              </View>
            </View>
            <View style={styles.mesSubContainer1}>
              <Text style={styles.text2}>评论： </Text>
              <Text style={styles.text3}>{item.content}</Text>
            </View>
          </View>
        </View>
      </TouchableOpacity>
    </View>
  );
}

praiseItemView = (item) => {
  return(<View><View>
    <TouchableOpacity onPress={()=>this.onClickPraise(item)}>
      <View style={styles.line}></View>
```

```
            <View style={styles.msgContainer}>
                <TouchableOpacity onPress={()=>this.onClickAvatar(item)}>
                    <Image source={{uri: item.to_user.avatar}} style={styles.avatar}/>
                    {/*<Image source={require('../source/未登陆.png')} style={styles.avatar}/>*/}
                </TouchableOpacity>
                <View style={styles.msgSubContainer}>
                    <View style={styles.mesSubContainer}>
                        <Text style={styles.text1}>{item.to_user.nick_name}</Text>
                    </View>
                    <View style={styles.mesSubContainer1}>
                        <Text style={styles.text2}>Ta给你点赞啦！ </Text>
                    </View>
                </View>
            </View>
        </TouchableOpacity>
    </View></View>);
}

chatItemView = (item) => {
    return(<View>
        <TouchableOpacity onPress={()=>this.onClickChat(item)}>
            <View style={styles.line}></View>
            <View style={styles.msgContainer}>
                <TouchableOpacity>
                    {/*<Image source={require('../source/未登陆.png')} style={styles.avatar}/>*/}
                    <Image source={{uri: item.to_user.avatar}} style={styles.avatar}/>
                </TouchableOpacity>
                <View style={styles.msgSubContainer}>
                    <View style={styles.mesSubContainer}>
                        <Text style={styles.text1}>{item.to_user.nick_name}</Text>
                        <View style={styles.container1}>
                            <Text style={styles.text3}>1小时前</Text>
                            <Text style={styles.text2}> * </Text>
                        </View>
                    </View>
                    <View style={styles.mesSubContainer1}>
                        <Text style={styles.text2}>私信： </Text>
                        <Text style={styles.text3}>{item.content}</Text>
                    </View>
                </View>
            </View>
        </TouchableOpacity>
    </View>)
}

noticeItemView = (item) => {

    return(<View><View>
        <TouchableOpacity onPress={()=>{
            alert(item.content);
        }
        }>
            <View style={styles.line}></View>
            <View style={styles.msgContainer}>
                <View style={styles.msgSubContainer}>
                    <View style={styles.mesSubContainer}>
                        <Text style={styles.text1}>消息通知</Text>
                        <View style={styles.container1}>
```

```
                                <Text style={styles.text3}>{item.updated_at}</Text>
                            </View>
                        </View>
                        <View style={styles.mesSubContainer1}>
                            <Text style={styles.text2}>{item.content}</Text>
                        </View>
                    </View>
                </View>
            </TouchableOpacity>
        </View></View>);
    }

}

const {width, height, scale} = Dimensions.get('window');

const styles = StyleSheet.create({
    container: {
        flex: 1,
    },
    topContainer: {
        paddingTop: 8,
        flexDirection: 'row',
        justifyContent: 'space-around',
        backgroundColor: 'white'
    },
    msgIcon: {
        width: 42,
        height: 42,
    },
    msgIconContainer: {
        alignItems: 'center',
        backgroundColor: 'white',
    },
    msgDes: {
        marginTop: 5,
        fontSize: 12,
    },
    msgContainer: {
        flexDirection: 'row',
        alignItems: 'center',
        height: 60,
    },
    avatar: {
        borderRadius: 19,
        width: 38,
        height: 38,
        marginLeft: 10,
    },
    line: {
        height: 0.5,
        marginLeft: 2,
        marginRight: 2,
        backgroundColor: gColor.grayLineColor,
    },
    mesSubContainer: {
        height: 30,
        flexDirection: 'row',
        justifyContent: 'space-between',
        alignItems: 'center',
        marginLeft: 10,
        marginRight: 10,
    },
    mesSubContainer1: {
        height: 14,
        flexDirection: 'row',
        alignItems: 'center',
        marginLeft: 10,
        marginRight: 10,
    },
    msgSubContainer: {
        flex: width - 74,
        height: 56,
    },
    container1: {
        flexDirection: 'row',
        alignItems: 'center',
        marginRight: 10,
    },
    text1: {
        fontSize: 14,
    },
    text2: {
        color: gColor.orangeTextColor,
    },
    text3: {
        color: '#333333',
    }
});
```

NewsDetail.js:

```
import React, { Component } from "react";
import {Text, View, StyleSheet, FlatList, TextInput, TouchableOpacity,
    Image, Dimensions, SectionList, SafeAreaView, Keyboard} from "react-native";
import httpApi from "../tools/Api";
import BannerView from "../component/BannerView"
import LoadingView from "../component/LoadingView";
import {gViewStyles} from "../style/ViewStyles";
import MyNavigationBar from '../component/MyNavigationBar'
import MyStatusBar from "../component/MyStatusBar";
import {gImageStyles} from "../style/ImageStyles";
import {gTextStyles} from "../style/TextStyles";

export default class NewsDetail extends Component {

    constructor(props) {
        super(props);
        this.state = {
            newsDetailData: [],
            commentList: [],
            load: false,
            toUserId: null,
            autoFocus: null,
            keyboardHeight: 0,
            commentId: null,
        };

        this.keyboardWillShowListener = null;
        this.keyboardWillHideListener = null;

    }

    async componentDidMount() {
        this.setState({
            newsDetailData: [],
            commentList: [],
            load: false,
            toUserId: null,
            autoFocus: null,
            keyboardHeight: 0,
            commentId: null,
        });
        let newsId = this.props.route.params.newsId;
        await this.fetchNewsData(newsId);
        await this.fetchNewsCommentListData(newsId);

        this.keyboardWillShowListener =
Keyboard.addListener('keyboardWillShow',this.keyboardDidShow);
        this.keyboardWillHideListener =
Keyboard.addListener('keyboardWillHide',this.keyboardDidHide);
    }

    //注销监听
    componentWillUnmount () {
        this.setState({
            newsDetailData: [],
            commentList: [],
            load: false,
            toUserId: null,
            autoFocus: null,
            keyboardHeight: 0,
            commentId: null,
        });
        this.keyboardWillShowListener && this.keyboardWillShowListener.remove();
        this.keyboardWillHideListener && this.keyboardWillHideListener.remove();
    }


    //键盘弹起后执行
    keyboardDidShow = (event) =>  {
        this.setState({
            keyboardHeight: event.endCoordinates.height
        });
    }

    //键盘收起后执行
    keyboardDidHide = () => {
        this.setState({
            keyboardHeight: 0
        });
    }

    // 点击广告
    onClickAd = (item) => {
        if (item.href != null && item.href.length > 0) {
            let params = {
                url: item.href,
            };
            this.props.navigation.navigate('WebPage', params);
        }
    }

    // 结束评论编辑
    onEndEditing = (text) => {
        this.uploadComment(text)
    }

    // 上传评论
    uploadComment = async (text) => {
        let params = '';
        params+=("post_id=" + this.props.route.params.newsId + '&');
        params+=('comment_content=' + text + '&');
        params+=('to_user_id=' + this.state.toUserId + '&');
        params+=('comment_id=' + this.state.commentId + '&');

        let res = await httpApi.httpPostWithParamsStr("http://dd.shenruxiang.com/api/v1/
send_post_comment", params);
        if (res.status == 0) {
            this.refs.textInputRefer.clear();
            let newsId = this.props.route.params.newsId;
            this.fetchNewsCommentListData(newsId);
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    // 请求帖子详情
    fetchNewsData = async (newsId) => {
        let params = {
            post_id: newsId
        };
        let res = await httpApi.getNewsData(params);

        if (res.status == 0) {

            this.setState({
                newsDetailData: res.data,
                toUserId: res.data.newsDetail.user.id,
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    onPraise = async (item) => {
        let params = '';
        params+=("post_id=" + this.props.route.params.newsId + '&');
        params+=('to_user_id=' + item.to_user_id + '&');
        params+=('comment_id=' + item.comment_id + '&');
        let res = await httpApi.httpPostWithParamsStr('http://dd.shenruxiang.com/api/v1/post_comment_praise', params);
        if (res.status == 0) {
            this.fetchNewsCommentListData(this.props.route.params.newsId);
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    // 请求帖子评论列表数据
    fetchNewsCommentListData = async (newsId) => {
        let params = {
            post_id: newsId
        };
        let res = await httpApi.getNewsCommentListData(params);

        if (res.status == 0) {
            let newCommentList = [];
            for (let i = 0; i < res.data.length; i++) {
                newCommentList.push({title:res.data[i], data:(res.data[i].reply_list != null ? res.data[i].reply_list : [])});
            }
            this.setState({
                commentList: newCommentList,
                load: true
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    render() {
        if (!this.state.load) {
            return <LoadingView/>;
        }
        return(
            <View style={gViewStyles.rootViewContainer}>
                <MyStatusBar/>
                <SafeAreaView style={styles.container}>
                    <MyNavigationBar
                        title={'帖子详情'}
                        onClickBack={()=>{this.props.navigation.goBack();}}
                    ></MyNavigationBar>
                    <View style={styles.topContainer}>
                        <SectionList
                            ListHeaderComponent={this.headerView}
                            sections={this.state.commentList}
                            renderItem={this.subCommentItemView}
                            renderSectionHeader={this.commentItemView}
                            stickySectionHeadersEnabled={false}
                            keyExtractor={(item, index) => item + index}
                        ></SectionList>
                    </View>

                    <View style={[styles.inputContainer, {marginBottom: this.state.keyboardHeight}]}>
                        <TextInput underlineColorAndroid="transparent"
```

```
                                    placeholder="输入评论内容"
                                    ref='textInputRefer'
                                    style={styles.input}
                                    returnKeyType="send"
                                    onSubmitEditing={(event)=>this.onEndEditing(event.nativeEvent.text)}>
                        </TextInput>
                        <TouchableOpacity>
                            <Image source={require('../source/xiangqingfenxiang.png')}
style={styles.shareIcon}/>
                        </TouchableOpacity>
                    </View>
                </SafeAreaView>
            </View>

        );
    };

    headerView = () => {
        return(
            <View style={styles.view2}>
                {this.userInfoItemView()}
                {this.newsInfoItemView()}
                <View style={styles.commentLine}></View>
                <BannerView
                    onClickAd = {this.onClickAd}
                    data={this.state.newsDetailData.adDetail.data}
                ></BannerView>
                {/*评论总数、点赞总数*/}
                <View style={styles.container7}>
                    <Text style={[{color: '#666666'}, ]}>全部评论：
{this.state.newsDetailData.newsDetail.comment_num}</Text>
                    <Text style={[{color: '#666666'}, ]}>点赞：
{this.state.newsDetailData.newsDetail.praise_num}</Text>
                </View>
                <View style={styles.bottomSegmentation}></View>
            </View>
        );
    }

    userInfoItemView = () => {
        return(
            <View style={styles.userInfoContainer}>
                <TouchableOpacity>
                    {/*<Image source={require('../source/未登陆.png')} style={styles.avatar}/>*/}
                    <Image source={{uri: this.state.newsDetailData.newsDetail.user.avatar}}
                            style={styles.avatar}/>
                </TouchableOpacity>
                <View style={styles.userInfoContainer1}>
                    <Text style={styles.userInfoText1}
>{this.state.newsDetailData.newsDetail.user.nick_name}</Text>
                    <Text style={styles.userInfoText}
>{this.state.newsDetailData.newsDetail.user.created_at} 来自：
                        {this.state.newsDetailData.newsDetail.user.phone_model}</Text>
                </View>
                {this.state.newsDetailData.newsDetail.user_follow == null ? (<TouchableOpacity
onPress={async ()=>{
                    let params = ('user_id=' + this.state.newsDetailData.newsDetail.user.id + '&');
                    let res = await httpApi.httpPostWithParamsStr('http://dd.shenruxiang.com/api/v1/
user_follow', params);
                    if (res.status = 0) {
                        let newsDetailData = this.state.newsDetailData;
                        newsDetailData.newsDetail.user_follow = 1;
                        this.setState({
                            newsDetailData: newsDetailData,
                        });
                        WToast.show({data: '关注成功!'});
                    } else {
                        alert("网络异常！请检查网络！");
                    }
                }}>
                    <View style={[gTextStyles.textBack, {marginRight: 5}]}>
                        <Text style={gTextStyles.text}> + 关注</Text>
                    </View>
                </TouchableOpacity>) : (<View></View>)}

            </View>
        );
    };

    newsInfoItemView = () => {
        return(
            <View style={styles.newsInfoContainer}>
                <View style={styles.commentLine}></View>
                <Text style={styles.newsInfoText}
>{this.state.newsDetailData.newsDetail.user.post_content}</Text>

                {/*动态图片部分*/}
                <FlatList
                    data={this.state.newsDetailData.newsDetail.source}
                    renderItem={this.imgItemView}
                    style={styles.newsImgList}
                />
                <View style={styles.newsInfoContainer1}>
                    <Image source={require('../source/dingwei1.png')} style={styles.itemIcon}/>
                    <Text>{this.state.newsDetailData.newsDetail.post_position}</Text>
                </View>
            </View>
        );
    };

    imgItemView({ item }) {
        return (
            <View>
                <Image source={{uri: item.src}} style={styles.newsImg}/>
                {/*<Image source={require('../source/banner.jpg')} style={gImageStyles.bannerImg}/>*/}
            </View>
        );
    }

    commentItemView = ({ section: { title, data } }) => {
        let commentData = title;

        return(
            <View>
                <TouchableOpacity onPress={()=>{
```

```
          this.setState({
            toUserId: commentData.user.id,
            commentId: commentData.id,
          });
          this.refs.textInputRefer.focus();
        }}>
          <View style={styles.commentContainer}>
            <Image source={{uri: commentData.user.avatar}} style={styles.img1}/>
              <View style={styles.commentContainer1}>
                <View style={styles.commentContainer2}>
                  <Text style={styles.commentText}>{commentData.user.nick_name}</Text>
                  <TouchableOpacity onPress={() => this.onPraise(commentData)}>
                    <View style={gViewStyles.praiseView}>
                      {commentData.comment_priase != null ? (
                        <Image source={require('../source/dianzanred.png')}
                           style={styles.commentImg}/>
                      ) : (<Image source={require('../source/dianzanblack.png')}
                            style={styles.commentImg}/>
                      )}
                      <Text style={styles.commentText2}>{commentData.praise_num}</
Text>
                    </View>
                  </TouchableOpacity>
                </View>
                <Text style={styles.commentText1}>{commentData.content}</Text>
                <Text style={styles.commentText3}>{commentData.created_at}</Text>
                <Text>回复Ta</Text>
              </View>
          </View>
          <View style={styles.commentLine}></View>
        </TouchableOpacity>
      </View>
    );
  };

  subCommentItemView = ({ item }) => {

    return(
      <View>
        <TouchableOpacity onPress={()=>{
          this.setState({
            toUserId: item.user.id,
            commentId: commentData.id,
          });
          this.refs.textInputRefer.focus();
        }}>
          <View style={styles.subCommentContainer}>
            <Image source={{uri: item.user.avatar}} style={styles.img1}/>
              <View style={styles.commentContainer3}>
                <View style={styles.commentContainer4}>
                  <Text style={styles.commentText}> {item.user.nick_name}</Text>
                  <TouchableOpacity onPress={() => this.onPraise(item)}>
                    <View style={gViewStyles.praiseView}>
                      {item.comment_priase != null ? (
                        <Image source={require('../source/dianzanred.png')}
                           style={styles.commentImg}/>
                      ) : (<Image source={require('../source/dianzanblack.png')}
                            style={styles.commentImg}/>
                      )}
```

```
                                            <Text style={styles.commentText2}>{item.praise_num}</Text>
                                        </View>
                                    </TouchableOpacity>
                                </View>
                                <Text style={styles.commentText1}>{item.user.nick_name} @
{item.to_user.nick_name}：{item.content}</Text>
                                <Text style={styles.commentText3}>{item.created_at}</Text>
                                <Text>回复Ta</Text>
                            </View>
                    </View>
                    <View style={styles.commentLine}></View>
                </TouchableOpacity>

        </View>);
    }
}

const {width, height, scale} = Dimensions.get('window');

const styles = StyleSheet.create({
    container: {
        flex: 1,
        backgroundColor: 'white',
    },
    topContainer: {
        flex: height - 40,
    },
    newsInfoList: {
    },
    inputContainer: {
        flexDirection: 'row',
        flex: 40,
        padding: 10,
        margin: 5,
        alignItems: 'center',
        backgroundColor: 'white',
    },
    input: {
        flex: width - 70,
        height: 40,
        borderWidth: 1,
        borderColor: 'black',
        padding: 5,
        borderRadius: 10,
    },
    shareIcon: {
        flex: 40,
        marginLeft: 10,
        color: 'white',
        width: 20,
        height: 20,
    },
    avatar: {
        flex: 60,
        height: 40,
        width: 40,
        margin: 5,
        borderRadius: 20,
    },
userInfoContainer: {
    margin: 5,
    flexDirection: 'row',
    alignItems: 'center',
},
userInfoContainer1: {
    flex: width - 130,
},
userInfoText: {
    margin: 5,
    fontSize: 12,
    color: '#666666'
},
userInfoText1: {
    margin: 5,
},
userInfoImg: {
    flex: 60,
    width: 60,
    height: 40,
    margin: 5,
},
newsInfoContainer: {
},
newsInfoLine1: {
    height: 1,
    backgroundColor: '#C0C0C0',
},
adContainer: {
    margin: 5,
},
adImg: {
    width: width - 10,
},
newsInfoLine: {
    height: 1,
    backgroundColor: "#C0C0C0",
},
newsInfoText: {
    margin: 10,
},
newsInfoContainer1: {
    margin: 5,
    flexDirection: 'row',
    alignItems: 'center',
},
commentLine: {
    margin: 5,
    height: 0.5,
    backgroundColor: "#D8D8D8",
},
commentContainer: {
    padding: 5,
    flexDirection: "row",
},
commentContainer1: {
    marginLeft: 5,
    marginRight: width - 70,
    justifyContent: 'center',
},
  commentText: {
    color: '#1A1A1A'
  },
  commentText1: {
    margin: 5,
    flex: width - 150,
    color: '#333333'
  },
  commentContainer2: {
    width: width - 70,
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'space-between'
  },
  commentContainer3: {
    marginLeft: 5,
    marginRight: width - 70 - 44,
    justifyContent: 'center',
  },
  commentContainer4: {
    width: width - 70 - 44,
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'space-between'
  },
  commentImg: {
    width: 30,
  },
  commentText2: {
  },
  commentText3: {
    color: "#C0C0C0",
  },
  subCommentContainer: {
    padding: 5,
    flexDirection: "row",
    marginLeft: 44,
  },
  view2: {
    flex: 1,
  },
  line4: {
    height: 0.5,
    backgroundColor: '#D8D8D8'
  },
  img1: {
    height: 40,
    width: 40,
    margin: 5,
    borderRadius: 20,
  },
  bottomSegmentation: {
    height: 10,
    backgroundColor: gColor.grayLineColor
  },
  text10: {
    backgroundColor: '#FB5442',
    color: 'white',
    fontSize: 12,
    padding: 8,
        height: 30,
        width: 60,
        margin: 5,
    },
    container7: {
        flexDirection: 'row',
        justifyContent: 'space-between',
        margin: 12,
    }
});
```

PersonalInfo.js:

```
import React, { Component } from "react";
import {Text, View, StyleSheet, TouchableOpacity, Image, FlatList, Dimensions, SectionList,
SafeAreaView} from "react-native";
import httpApi from "../tools/Api";
import LoadingView from "../component/LoadingView";
import {gViewStyles} from "../style/ViewStyles";
import MyNavigationBar from '../component/MyNavigationBar'
import MyStatusBar from "../component/MyStatusBar";
import NewsView from "../component/news/NewsView"

// 个人详情页，注意与my页区分
// 需要传入userid
export default class PersonInfo extends Component {

    constructor(props) {
        super(props);
        this.state = {
            personalData: [],
            albumData: [],
            attentionData: [],
            load: false,
            showType: 0, // 展示数据类型，0：资料部分，1：相册 2: 关注
            userId: null,
        };
    }

    componentDidMount() {
        this.setState({
            userId: this.props.route.params.userId
        });
        this.showNewsList();
    }

    showNewsList = () => {
        this.setState({
            load: false,
            showType: 0,
        });
        this.fetchPersonalData(this.state.userId);
    }

    showalbum = () => {
        this.setState({
            load: false,
            showType: 1,
        });
        this.fetchAlbumData(this.state.userId);
    }

    showAttentioin = () => {
        this.fetchAttentionData();
    }

    // 点击关注
    onClickAttention = (item) => {
        // 跳转个人详情
        this.setState({
            userId: item.to_user_id,
            personalData: [],
            albumData: [],
            attentionData: [],
        });
        this.showNewsList();
    }

    // 请求个人数据
    fetchPersonalData = async (userId) => {
        let params = {
            user_id: userId,
        };
        let res = await httpApi.getPersonalData(params);

        if (res.status == 0) {

            this.setState({
                personalData: res.data,
                load: true
            });
        } else {
            alert("网络异常！ 请检查网络！ ");
        }
    }

    // 请求相册数据
    fetchAlbumData = async (userId) => {
        let params = {
            user_id: userId,
        };
        let res = await httpApi.getAlbumData(params);

        if (res.status == 0) {
            let newAlbumArr = [];
            for (var key in res.data) {
                var value = res.data[key];
                let subAlbumArr = value;
                let dataArr = [{data:subAlbumArr}];
                newAlbumArr.push({title:key, data:dataArr});
            }
            this.setState({
                albumData: newAlbumArr,
                load: true
            });
    } else {
      alert("网络异常！请检查网络！");
    }
  }

  // 请求关注列表
  fetchAttentionData = async() => {
    let params = {

    };
    let res = await httpApi.httpPost("http://dd.shenruxiang.com/api/v1/user_follow_list",
params);
    if (res.status == 0) {
      this.setState({
        attentionData: res.data,
        showType: 2,
        load: true
      });
    } else {
      alert("网络异常！请检查网络！");
    }
  }

  render() {
    if (this.state.load == false) {
      return <LoadingView></LoadingView>
    }

    return(
      <View style={{ flex: 1, backgroundColor: 'white' }}>
        <MyStatusBar/>
        <SafeAreaView style={gViewStyles.rootViewContainer}>
          <MyNavigationBar
            title={'个人信息'}
            onClickBack={()=>{this.props.navigation.goBack();}}
          ></MyNavigationBar>

          <View style={styles.itemContainer1}>
            {/*有头像的部分*/}
            <View style={styles.container2}>
              <Image source={{uri: this.state.personalData.user_info.avatar}}
                   style={styles.img1}/>
              <View style={styles.container3}>
                <Text style={styles.text1}>{this.state.personalData.user_info.nick_name}</
Text>
                <Text style={styles.text2}>签名：{this.state.personalData.user_info.desc}</
Text>
                <Text style={styles.text3}>粉丝：{this.state.personalData.user_info.fan_num}
关注：{this.state.personalData.user_info.follow_num}</Text>
              </View>
            </View>
            <View style={styles.container4}>
              <TouchableOpacity onPress={this.showNewsList}>
                <Text style={styles.text4}>资料</Text>
                {this.state.showType == 0 ? (<View style={gViewStyles.scrollBarBottomLine2}
></View>) : (<View></View>)}
```

```
                            </TouchableOpacity>
                            <TouchableOpacity onPress={this.showalbum}>
                                <Text style={styles.text4}>相册</Text>
                                {this.state.showType == 1 ? (<View style={gViewStyles.scrollBarBottomLine2}
></View>) : (<View></View>)}
                            </TouchableOpacity>
                            {/*<TouchableOpacity onPress={this.showAttentioin}>*/}
                                {/*<Text style={styles.text4}>关注</Text>*/}
                                {/*{this.state.showType == 2 ? (<View
style={gViewStyles.scrollBarBottomLine2}></View>) : (<View></View>)}*/}
                            {/*</TouchableOpacity>*/}
                        </View>
                        <View style={styles.line5}></View>
                    </View>

                    {this.listView()}

                    <View style={styles.container6}>
                        <Image source={require('../source/privateChat.jpg')} style={styles.img2}/>
                        <Image source={require('../source/attention.jpg')} style={styles.img2}/>
                    </View>
                </SafeAreaView>
            </View>


        );
    };

    listView = () => {
        if (this.state.showType == 0) {
            // 资料
            return(
                <FlatList
                    extraData={this.state}
                    style={styles.container}
                    data={this.state.personalData.post_list}
                    renderItem={({item})=>(<NewsView
                        onClickNews={()=>{
                            let params = {
                                newsId: item.id,
                            };
                            this.props.navigation.navigate('NewsDetail', params);
                        }}
                        data={item}
                    ></NewsView>)}
                    ListHeaderComponent={this.topView}
                />
            );
        } else if (this.state.showType == 1) {
            // 相册
            return(<SectionList
                style={styles.sectionList1}
                stickySectionHeadersEnabled={false}
                ListHeaderComponent={this.topView}
                sections={this.state.albumData}
                renderItem={this.albumItemView}
                renderSectionHeader={this.albumHeaderItemView}
                renderSectionFooter={()=><View style={styles.line3}></View>}
            ></SectionList>);
        } else {
            // 关注
            return(<FlatList
                extraData={this.state}
                data={this.state.attentionData}
                renderItem={this.attentionItemView}
                style={gViewStyles.flatList}
            />);
        }
    }

    topView = () => {
        return(
            <View style={gViewStyles.viewContainer2}>

                {
                    this.state.showType != 2 ? (<View style={styles.container5}>
                        <Text style={styles.text5}>性别：{this.state.personalData.user_info.sex}</Text>
                        <Text style={styles.text5}>星座：摩羯座</Text>
                        <Text style={styles.text5}>感情状况：单身</Text>
                        <Text style={styles.text5}>出生日期：2000-01-01</Text>
                        <Text style={styles.text5}>职业：互联网-电子工程</Text>
                        <Text style={styles.text5}>注册地点：{this.state.personalData.user_info.province}-
{this.state.personalData.user_info.city}</Text>
                    </View>) : (<View></View>)
                }

            </View>
        );
    }

    attentionItemView = ({item}) => {
        return(
            <View>
                <TouchableOpacity onPress={()=>this.onClickAttention(item)}>
                    <View style={styles.line}></View>
                    <View style={styles.msgContainer}>
                        <Image source={{uri: item.to_user.avatar}} style={styles.avatar}/>
                        <View style={styles.msgSubContainer}>
                            <View style={styles.mesSubContainer}>
                                <Text style={styles.attentionNickNametext}>{item.to_user.nick_name}</Text>
                            </View>
                            <View style={styles.mesSubContainer1}>
                                <Text style={styles.attentiontext1}>你关注了Ta！</Text>
                            </View>
                        </View>
                    </View>
                </TouchableOpacity>
            </View>
        );
    }

    newsItemView = ({ item }) => {
        return(
            <View style={styles.itemContainer2}>
                <Text style={styles.text5}>{item.created_at}</Text>
                <View style={styles.line2}></View>
                <Text style={styles.text6}>{item.post_content}</Text>
                {/*<Image source={{uri: item.src}} style={styles.newsImg}/>*/}
                {/*<Image source={require('../source/广告4.jpg')} style={styles.newsImg}/>*/}

                {/*动态图片部分*/}
                <FlatList
                    data={item.source}
                    renderItem={this.newsImgItemView}
                    style={styles.newsImgList}
                    numColumns={2}
                />
                <View style={styles.bottomContainer}>
                    <View style={styles.bottomTopContainer}>
                        <Image source={require('../source/dingwei1.png')} style={styles.itemIcon}/>
                        <Text>{item.post_position}</Text>
                    </View>
                    <View style={styles.segmentation}></View>
                    <View style={styles.bottomBottomContainer}>
                        <View style={styles.bottomBottomSubContainer}>
                            <Image source={require('../source/fenxiang.png')} style={styles.itemIcon}/>
                            <Text>{item.share_num}</Text>
                        </View>
                        <View style={styles.bottomBottomSubContainer}>
                            <Image source={require('../source/pinglun1.png')} style={styles.itemIcon}/>
                            <Text>{item.comment_num}</Text>
                        </View>
                        <TouchableOpacity onPress={()=>this.onPraise(item)}>
                            <View style={styles.bottomBottomSubContainer}>
                                <Image source={require('../source/dianzanblack.png')} style={styles.itemIcon}/
>
                                <Text>{item.praise_num}</Text>
                            </View>
                        </TouchableOpacity>
                    </View>
                </View>
            </View>
        );
    };

    albumItemView = ({item}) => {
        return (
            <View style={styles.view1}>
                <FlatList
                    data={item.data}
                    renderItem={this.imgItemView}
                    numColumns={3}
                >
                </FlatList>
            </View>

        );
    }

    imgItemView = ({item}) => {
        return(
            <View>
```

```
                <Image source={{uri: item.src}} style={styles.albumImg}/>
            </View>
        );
    }

    albumHeaderItemView = ({section}) => {
        return(
            <View>
                <Text style={styles.text8}>{section.title}</Text>
            </View>
        );
    }

    newsImgItemView({ item }) {
        return (
            <View>
                <Image source={{uri: item.src}} style={styles.newsImgs}/>
            </View>
        );
    }
}

const {width, height, scale} = Dimensions.get('window');

const styles = StyleSheet.create({
    superContainer: {
        flex: 1,
    },
    container: {
        // backgroundColor: '#FB5442',
    },
    itemContainer1: {
        backgroundColor: '#FB5442',
    },
    container2: {
        flexDirection: 'row',
        alignItems: 'center',
    },
    img1: {
        height: 50,
        width: 50,
        margin: 10,
        borderRadius: 25,
    },
    container3: {
        margin: 10,
    },
    text1: {
        color: 'white',
        fontSize: 18,
    },
    text2: {
        color: 'white',
        marginTop: 5,
        marginBottom: 5,
    },
    text3: {
        color: 'white',
    },
    container4: {
        padding: 5,
        flexDirection: 'row',
        backgroundColor: 'white',
        alignItems: 'center',
    },
    text4: {
        margin: 5,
        fontSize: 17,
    },
    container5: {
        marginTop: 1,
        backgroundColor: 'white',
        padding: 10,
    },
    itemContainer2: {
        marginTop: 5,
        backgroundColor: 'white',
    },
    text6: {
        padding: 10,
    },
    line2: {
        height: 1,
        backgroundColor: gColor.grayLineColor,
    },
    bottomTopContainer: {
        marginTop: 5,
        height: 30,
        marginLeft: 5,
        flexDirection: 'row',
        alignItems: 'center'
    },
    segmentation: {
        marginTop: 5,
        marginLeft: 10,
        marginRight: 10,
        height: 1,
        backgroundColor: gColor.grayLineColor
    },
    bottomBottomContainer: {
        height: 40,
        marginLeft: 10,
        marginRight: 10,
        marginTop: 5,
        marginBottom: 5,
        flexDirection: 'row',
        alignItems: 'center',
        justifyContent: 'space-between'
    },
    itemIcon: {
        marginTop: 2,
        marginBottom: 2,
        marginLeft: 2,
        marginRight: 2,
        height: 30,
        width: 30,
    },
    bottomBottomSubContainer: {
        flexDirection: 'row',
        alignItems: 'center',
    },
    img2: {
        height: 30,
        width: width/2 - 100,
    },
    container6: {
        padding: 10,
        backgroundColor: 'white',
        height: 40,
        flexDirection: 'row',
        justifyContent: 'space-around',
        alignItems: 'center',
    },
    line3: {
        height: 10,
        backgroundColor: '#F2F2F2'
    },
    line5: {
        height: 0.5,
        backgroundColor: '#F2F2F2',
    },
    sectionList1: {
    },
    text8: {
        color: '#333333',
        fontSize: 11.5,
        marginTop: 10,
        marginLeft: 11,
    },
    view1: {
        flex: 1,
    },
    newsImg: {
        width: (gScreen.screen_width - 20) / 2,
        height: (gScreen.screen_width - 20) / 2,
        margin: 5,
        borderRadius: 5,
    },
    albumImg: {
        width: (gScreen.screen_width - 30) / 3,
        height: (gScreen.screen_width - 30) / 3,
        margin: 5,
        borderRadius: 5,
    },
    attentionNickNametext: {
        fontSize: 17,
    },
    attentiontext1: {
        color: gColor.orangeTextColor,
    },
    msgContainer: {
        flexDirection: 'row',
        alignItems: 'center',
        height: 60,
    },
    avatar: {
        borderRadius: 27,
        width: 54,
        height: 54,
        marginLeft: 10,
    },
    msgSubContainer: {
        width: width - 100,
        marginLeft: 7,
    },
    newsImgs: {
        marginTop: 2,
        marginBottom: 2,
        marginLeft: 2,
        marginRight: 2,
        width: width/2 - 5,
        height: width/2 - 5,
    },
});
```

Search.js:

```
import React, { Component } from "react";
import { FlatList, View, TouchableOpacity, Image, TextInput, SafeAreaView } from "react-native";
import {gImageStyles} from "../style/ImageStyles";
import {gViewStyles} from "../style/ViewStyles";
import MyStatusBar from "../component/MyStatusBar";
import {gTextStyles} from "../style/TextStyles";
import BannerView from "../component/BannerView"
import NewsView from "../component/news/NewsView"
import httpApi from "../tools/Api";
import LoadingView from "../component/LoadingView";

export default class Search extends Component {

    constructor(props) {
        super(props);
        this.state = {
            searchList: [],
            load: false
        };
    }

    componentDidMount() {
        this.fetchSearchData(this.props.route.params.search)
    }

    onEndEditing = (text) => {
        this.fetchSearchData(text)
    }

    // 搜索接口
    fetchSearchData = async (text) => {
        let params = {
            search: text,
        };
        let res = await httpApi.getNewsList(null);

        if (res.status == 0) {

            this.setState({
                searchList: res.data.data,
                load: true
            });
        } else {
            alert("网络异常！请检查网络！");
        }
    }

    render() {

        if (!this.state.load) {
            return <LoadingView/>;
        }

        return (
            <View style={gViewStyles.rootViewContainer}>
                <MyStatusBar></MyStatusBar>
                <SafeAreaView style={gViewStyles.rootViewContainer}>
                    <View style={gViewStyles.view3}>
                        {/*返回*/}
                        <TouchableOpacity onPress={()=>{this.props.navigation.goBack();}}>
                            <View style={gViewStyles.backView}>
                                <Image source={require('../source/fanhui.png')}
style={gImageStyles.leftArrow}/>
                            </View>
                        </TouchableOpacity>
                        {/*搜索框*/}
                        <View style={gViewStyles.searchView}>
                            <Image source={require('../source/sousuo.png')} style={gImageStyles.avatar}/>
                            <TextInput underlineColorAndroid="transparent" placeholder="搜索你想要的内
容" placeholderTextColor={'#999999'}
                                       style={gTextStyles.searchTextInput}
                                       onSubmitEditing={(event)=>this.onEndEditing(event.nativeEvent.text)}>
                                {this.props.route.params.search}
                            </TextInput>
                        </View>
                    </View>
                    {/*搜索结果列表*/}
                    <FlatList
                        data={this.state.searchList}
                        renderItem={this.newsListItemView}
                    />
                </SafeAreaView>
            </View>
        );
    }

    newsListItemView = (item) => {
        if (item.item.is_adv) {
            return(<BannerView
                onClickAd = {this.onClickAd}
                data={item.item.data}
            ></BannerView>);
        } else {
            return (
                <NewsView
                    onClickAvatar = {this.onClickAvatar}
                    data={item.item.data}
                    onClickNews={this.onClickNews}
                    isShowPersonInfo={true}
                ></NewsView>
            );
        }

    };
}
```

Type.js:

```
import React, { Component } from "react";
import { Text, View, SafeAreaView, FlatList, TouchableOpacity, DeviceEventEmitter } from "react-native";
import MyStatusBar from '../component/MyStatusBar'
import MyNavigationBar from '../component/MyNavigationBar'
import {gViewStyles} from "../style/ViewStyles";
import httpApi from "../tools/Api";
import {gTextStyles} from "../style/TextStyles";

export default class Type extends Component {

  constructor(props) {
    super(props);
    this.state = {
      data: [], // 动态分类数据
    };
  }

  componentDidMount() {
    this.fetchTypeData();
  }

  fetchTypeData = async () => {
    let res = await httpApi.httpPostWithParamsStr("http://dd.shenruxiang.com/api/v1/post_cate_list", null);
    if (res.status == 0) {
      this.setState({
        data: res.data,
      });
    } else {
      alert("网络异常！请检查网络！");
    }
  }

  onClickItem = (item) => {
    //调用事件通知  param是指传递的相应参数
    DeviceEventEmitter.emit('updateType', item);
    this.props.navigation.goBack();
  }

  render() {
    return (
      <View style={gViewStyles.rootViewContainer}>
        <MyStatusBar></MyStatusBar>
        <SafeAreaView>
          <MyNavigationBar
            title={'动态分类'}
```

```
                        onClickBack={()=>{this.props.navigation.goBack();}}
                    ></MyNavigationBar>
                    <FlatList
                        extraData={this.state}
                        data={this.state.data}
                        renderItem={({item})=>(
                            <TouchableOpacity onPress={()=>this.onClickItem(item)}>
                                <View style={gViewStyles.line}></View>
                                <View style={gViewStyles.itemView}>
                                    <Text style={gTextStyles.nickNameText}>{item.name}</Text>
                                </View>
                                <View style={gViewStyles.segmentation}></View>
                            </TouchableOpacity>
                            )}
                        style={gViewStyles.flatList}
                    />
                </SafeAreaView>
            </View>

        );
    }
}
```

WebPage.js:

```
import React, { Component } from "react";
import { Text, View, SafeAreaView } from "react-native";
import MyStatusBar from '../component/MyStatusBar'

export default class WebPage extends Component {

    render() {
        return (
            <View>
                <MyStatusBar></MyStatusBar>
                <SafeAreaView>
                    <WebView
                        source={{uri: this.props.route.params.url}}
                    />
                </SafeAreaView>
            </View>

        );
    }
}
```

Api.js:

```
import http from './Http';

// promise的封装操作
function to(promise){
    return promise
        .then(data => {
            return [null, data];
        })
        .catch(err => [err]);
}
```

```
const httpApi={

    async httpPost(url, params) {
        const [err, res] = await to(http.post(url, params));
        if (err) {
            //请求失败
            return Object.assign(err, {
                status: "406",
                description: err.errMsg
            }, true);
        }
        //请求成功
        return res;
    },

    async httpPostWithParamsStr(url, paramsStr) {
        const [err, res] = await to(http.postWithStr(url, paramsStr));
        if (err) {
            //请求失败
            return Object.assign(err, {
                status: "406",
                description: err.errMsg
            }, true);
        }
        //请求成功
        return res;
    },

    // 获取新闻分类列表
    async getNewsTypeList(params) {
        const url = 'http://dd.shenruxiang.com/api/v1/post_cate_list';
        const [err, res] = await to(http.post(url, params));
        if (err) {
            //请求失败
            return Object.assign(err, {
                status: "406",
                description: err.errMsg
            }, true);
        }
        //请求成功
        return res;
    },


}
export default httpApi;
```