

1. rn flatlist 渲染问题

- 对于除data以外的数据需要添加extraData
- 若data和extraData中都是数组，修改该数据，需要同时修改其引用地址，例如重新赋值一个新数组

```
1      return;  
2    }  
3    const newData = lodash.remove(hiredData,  
4      item => item.userLoginName !== currentObj.userLoginName);  
5    this.setState({ hiredData: newData });  
6    Toast.info('删除成功', 1.5);
```

2. rn 从A页面路由到B页面，再返回A页面，触发更新方案

- 在A页面添加监听事件，卸载时移除，在B页面goback（）之前emit派发该事件

```
componentWillMount() {  
  const { navigation } = this.props;  
  navigation.setParams({ onPressAddBtn: this.toAddEmployee });  
  this.onRefresh();  
  // 注册监听事件  
  this.listener = DeviceEventEmitter.addListener(DeviceEventName.STAFF_MANAGE, () => {  
    this.onRefresh();  
  });  
}  
  
componentWillUnmount() {  
  // 移除通知监听事件(移除之前先判空处理，增强代码强壮性)  
  if (this.listener) {  
    this.listener.remove();  
  }  
}
```

```
323      Toast.info('添加员工成功', 1.5);  
324      setTimeout(() => {  
325        Toast.hide();  
326 +      DeviceEventEmitter.emit(DeviceEventName.STAFF_MANAGE);  
327 +      navigation.goBack();  
328      }, 1500);  
---
```

3. rn 多个text并写，会有样式不对称问题，可以用view隔开处理

```

<TouchableOpacity
  activeOpacity={0.8}
  style={styles.supplierContent}
  onPress={() => {
    onChangeSupplier('custom');
  }}
>
  <Text style={styles.supplierText}>已指定</Text>
  <View><Text style={styles.supplierNumberText}>{supplierNumber}</Text></View>
  <Text style={styles.supplierText}>家供应商</Text>
  <View style={styles.arrowRightContainer}>

```

4. DeviceEventEmitter

```

componentDidMount() {
  this.appearListener = DeviceEventEmitter.addListener('navigation_appear', async ({ moduleName }) => {
    if (moduleName && moduleName === 'ContractSigning') {
      await this.loadPage();
    }
  });
}

componentWillUnmount() {
  if (this.appearListener) {
    this.appearListener.remove();
  }
}

```

5. 设置opacity, 子view会继承, 设置如果把就可以避免这个问题

6. Image borderTopLeftRadius等属性实现部分圆角会不起作用

- 用imagebackground组件, 并在style里面使用圆角和属性overflow: 'hidden'

7. debounce(防抖)和throttle(节流)用来控制某个函数在一定时间内执行多少次的技巧, 输入框触发函数常见

```

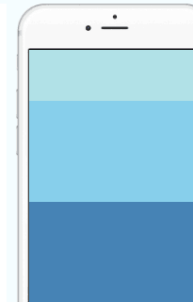
historyVin.map(item => (
  <TouchableHighlight
    underlayColor="#e6e6e6"
    onPress={debounce(() => this.pressItem(item.vinCode), 500)}
    key={item.vinCode}
  >
    <View style={styles.itemContainer}>

```

8. 子 view撑不开

组件能够撑满剩余空间的前提是其父容器的尺寸不为零。如果父容器既没有固定的 `width` 和 `height`，也没有设定 `flex`，则父容器的尺寸为零。其子组件如果使用了 `flex`，也是无法显示的。

```
1 import React, { Component } from 'react';
2 import { AppRegistry, View } from 'react-native';
3
4 export default class FlexDimensionsBasics extends Component {
5   render() {
6     return (
7       // 试试去掉父View中的`flex: 1`。
8       // 则父View不再具有尺寸，因此子组件也无法再撑开。
9       // 然后再用`height: 300`来代替父View的`flex: 1`试试看？
10      <View style={{flex: 1}}>
11        <View style={{flex: 1, backgroundColor: 'powderblue'}} />
12        <View style={{flex: 2, backgroundColor: 'skyblue'}} />
13        <View style={{flex: 3, backgroundColor: 'steelblue'}} />
14      </View>
15    );
16  }
```



9. 出现过好几次白屏问题了，如果data 为空，这个View下面就是{},android 会白屏的

```
95   buttonText = '立即支付';
96 }
97 return (
98   <TouchableOpacity
99     onPress={() => this.openOrderDetail(orderInfo.orderId)}
100     activeOpacity={1}
101     style={styles.container}
102   >
103     <View style={styles.orderView}>
104       <Text style={styles.orderId}>{orderInfo.orderId}</Text>
105       <Text style={styles.orderStatus}>{OrderStatusText[orderInfo.orderStatus]}</Text>
106     </View>
107     <ScrollView contentContainerStyle={styles.scrollView}>
108       {data.map((item, index) => this.renderItem(item, index))}
109     </ScrollView>
110     <View style={styles.priceView}>
111       <Text style={styles.priceDescribe}>{'应付: '}</Text>
112       <Text style={styles.price}>{orderInfo.total}</Text>
113       <Text style={styles.priceDescribe}>元</Text>
114     </View>
115     <KsButton
116       type="primary"
117       text={buttonText}
118       shadow={true}
119       onPress={() => this.openOrderDetail(orderInfo.orderId)}
120       style={styles.button}
121     />
122   </TouchableOpacity>
123 )
```

10. connect绑定的子组件，父组件引用该子组件，使用ref调子组件的函数会报找不到，需要给connect添加{ withRef: true }

```
123
124 TopMenu.propTypes = propTypes;
125
126 const mapDispatchToProps = dispatch => ({
127   currentActions: bindActionCreators(currentCreators, dispatch),
128   dispatch
129 });
130
131 export default connect(null, mapDispatchToProps, [], { withRef: true })(TopMenu);
132
```

```
if (isLoadingDataError) {
  return (
    <SafeAreaContainer>
      <View style={styles.container}>
        { /* 顶部菜单 */ }
        <TopMenu { /* */ }
          ref={ (ref) => { this.topMenu = ref; } }
          navigation={navigation}
        />
        { /* 首页banner 部分 */ }
        <ImageBackground
          source={this.renderStoreBg()}
          style={styles.HomeBanner}
        />
      </View>
    </SafeAreaContainer>
  );
}
```

```
onShowTopMenu() {
  this.topMenu.toShowHideModal();
}

/**
```

11. 使用flatlist,或者sectionlist的时候，经常会遇到视图不更新的情况，

- 本组件继承自PureComponent而非普通的Component，这意味着如果其props在浅比较中是相等的，则不会重新渲染。所以请先检查你的renderItem函数所依赖的props数据（包括data属性以及可能用到的父组件的state），如果是一个引用类型（Object或者数组都是引用类型），则需要先修改其引用地址（比如先复制到一个新的Object或者数组中），然后再修改其值，否则界面很可能不会刷新。（译注：这一段不了解的朋友建议先学习下js中的基本类型和引用类型。）

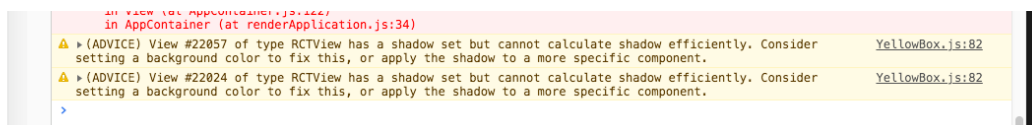
React.PureComponent实现了props和state进行浅比较的shouldComponentUpdate()方法。如果你确定想人肉处理这个浅比较操作，你可以自己在这个函数中比较this.props和nextProps、this.state和nextState是否相同。相同返回false，不同返回true，那么React就会根据返回值来确认是否跳过本次DOM渲染。

所以，当state内存放的数据为数组时候，实质上是一个引用，就算是state内push了一个数据，这时候是不会触发更新的，

我这里这么解决的

```
1 let d = {
2     nr:action.mes,
3     type:"txt",
4     who:1
5 };
6
7 list = [];
8 list = list.concat(this.state.Data);
9 list.push(d);
```

12. 给view加阴影时使用shadow,会黄色警告，需要增加backgroundColor解决



13. flatlist 下拉刷新触发多次上拉加载方法，并且上拉加载也会触发多次，
解决方案：定义一个变量限制上拉加载方法请求多次，并且添加
onEndReachedThreshold 和 onMomentumScrollBegin 属性

```
<StatusBar hidden={false} backgroundColor={ '#fff' } />
<FlatList
  data={this.props.notificationList}
  renderItem={this.getItem}
  refreshing={this.state.isLoading}
  onRefresh={() => this.onRefresh()}
  keyExtractor={this.getItemKey}
  onEndReached={() => this.onEndReached()}
  onMomentumScrollBegin={() => {
    this.onEndReachedCalled = false;
  }}
  onEndReachedThreshold={0.2}
/>
</SafeAreaContainer>
);
```

```
async onEndReached() {
  const { user, notificationActions, notificationList } = this.props;
  if (this.onEndReachedCalled) {
    return;
  }
  const lastId = notificationList.length ? last(notificationList)._id : 0; // eslint-disable
  try {
    const result = await notificationActions.getAndUpdateNotificationList({
      platform: Platform.OS,
      to: user.userLoginId,
      beginId: lastId
    }, true);

    if (result.errorCode > 0) { Toast.info(result.message, 1.5); }
    this.onEndReachedCalled = true;
  } catch (e) {
    handleHttpError(e);
  }
}
```

You, 11 minutes ago | 7 authors (Kai_Lin and others)

```
class NotificationScreen extends PureComponent {
  constructor(props) {
    super(props);
    this.onEndReachedCalled = false;
    this.state = {
      isLoading: false
    };
    this.getItem = this.getItem.bind(this);
    this.getItemKey = this.getItemKey.bind(this);
    this.onRefresh = this.onRefresh.bind(this);
    this.getNotifications = this.getNotifications.bind(this);
    this.onEndReached = this.onEndReached.bind(this);
  }
}
```

14. Text显示数字加粗后最后一位无法显示问题，即设置过fontWeight属性的Text，在OPPO等机型确实有问题

解决办法：给Text加个fontFamily样式：fontFamily: 'System'

15.Text显示金额的时候字体加粗后显示不全的问题，即设置过fontWeight属性的Text，在OPPO等机型确实有问题

解决办法：给Text加个fontFamily样式：fontFamily: 'System' 或者 fontFamily: ''