

02

OPEN ORIENTED

凹凸实验室

React Native 入门实战

阮嘉伟

0 项目介绍

1 结构分析

2 涉及知识点

3 一些坑整理

4 项目总结

0

项目介绍

京东秒杀

线上地址：

<https://ms.m.jd.com/seckill/seckillList>

京东秒杀

10:00
秒杀中

12:00
即将开场

14:00
即将开场

16:00
即将开场

18:00
即将开场

即将开场 先下单先得哦

距开始 01:07:01

品质



小米 (MI) 路由器3 1200M千兆5G双频
四天线无线路由 智能wifi稳定穿墙高速...

¥139
¥169

提醒我

大牌秒



联想(Lenovo)小新出色版510S 14英寸轻
薄笔记本电脑 (I7-7500U 8G 256SS...

¥4799
¥4999

提醒我



美的 (Midea)电饭煲WFZ4023XM IH电
磁加热 WiFi操控 精钢鼎釜4L电饭锅

¥599
¥999

提醒我



【前500单仅84】滴露消毒液1.5*2瓶特
惠装 家居洗衣宠物地板除菌液可搭配...

¥89

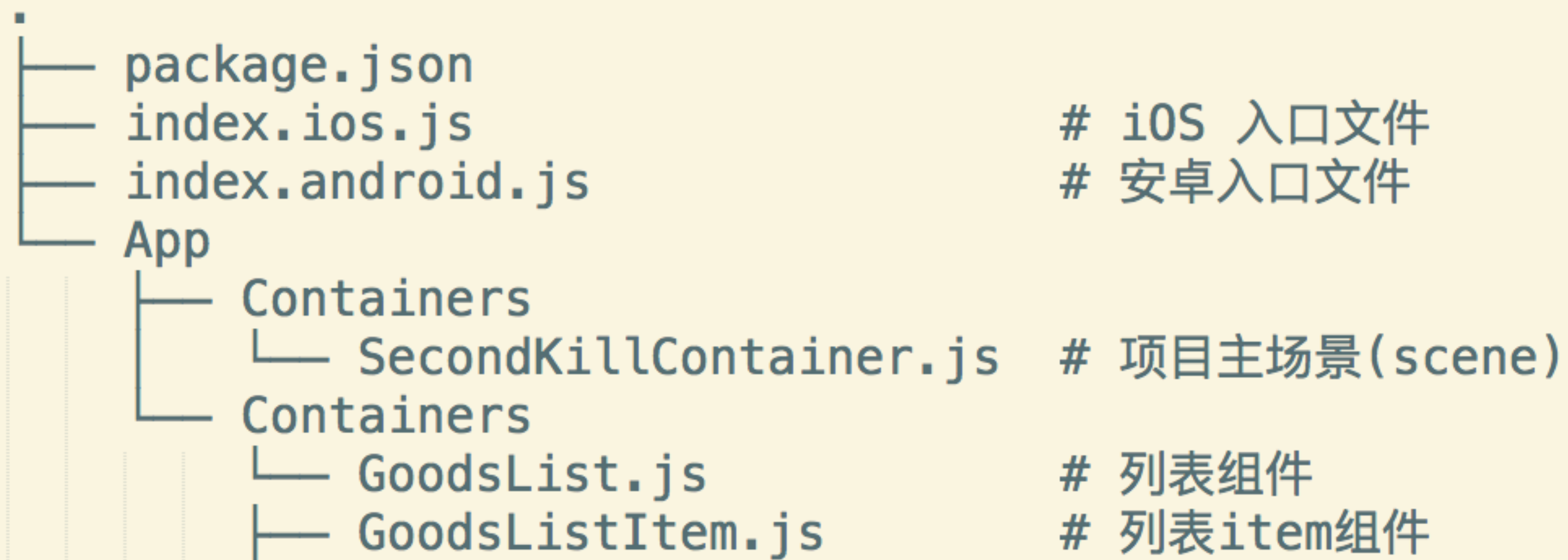
提醒我

京品秒杀

品牌秒杀

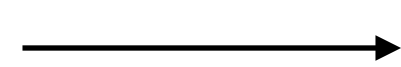
2

项目结构



APP 启动

index.android.js
index.ios.js



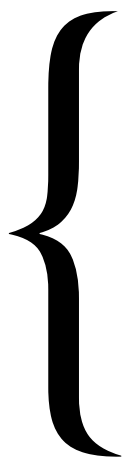
主场景

App/Containers/**SecondKillContainer.js**

主场景

包含的组件

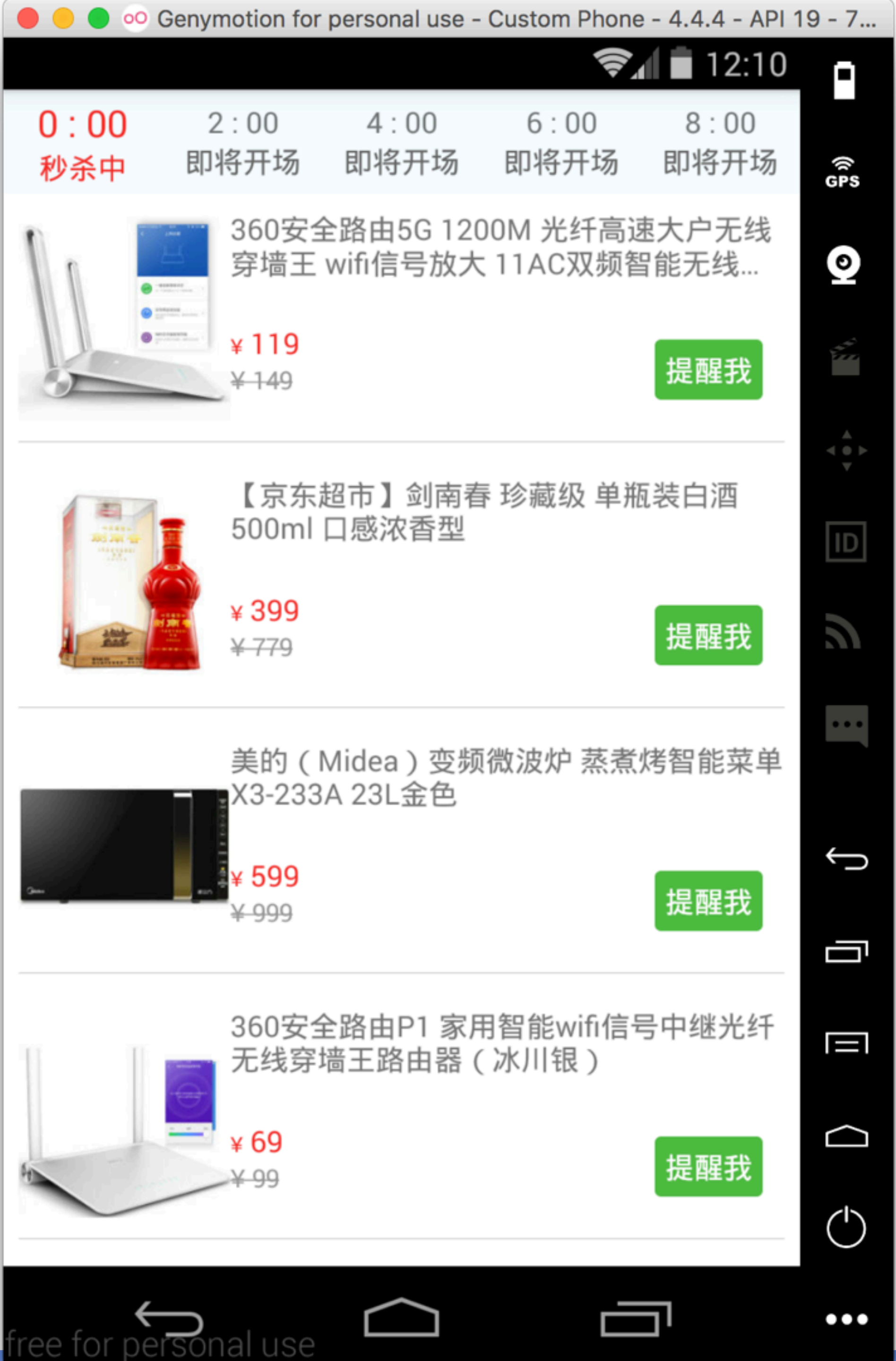
→ App/Containers/**SecondKillContainer.js**



App/Components/**GoodsList.js**

App/Components/**GoodsListItem.js**

项目视图结构





TabsView

View 组件 + Flexbox 布局

GoodsList

Listview 组件

1

涉及知识点

ES6

UI布局

网络请求

调试

代码逻辑

数据流

- ◆ Class
- ◆ 箭头函数
- ◆ 解构赋值
- ◆ import / export
- ◆ 参数默认值
- ◆

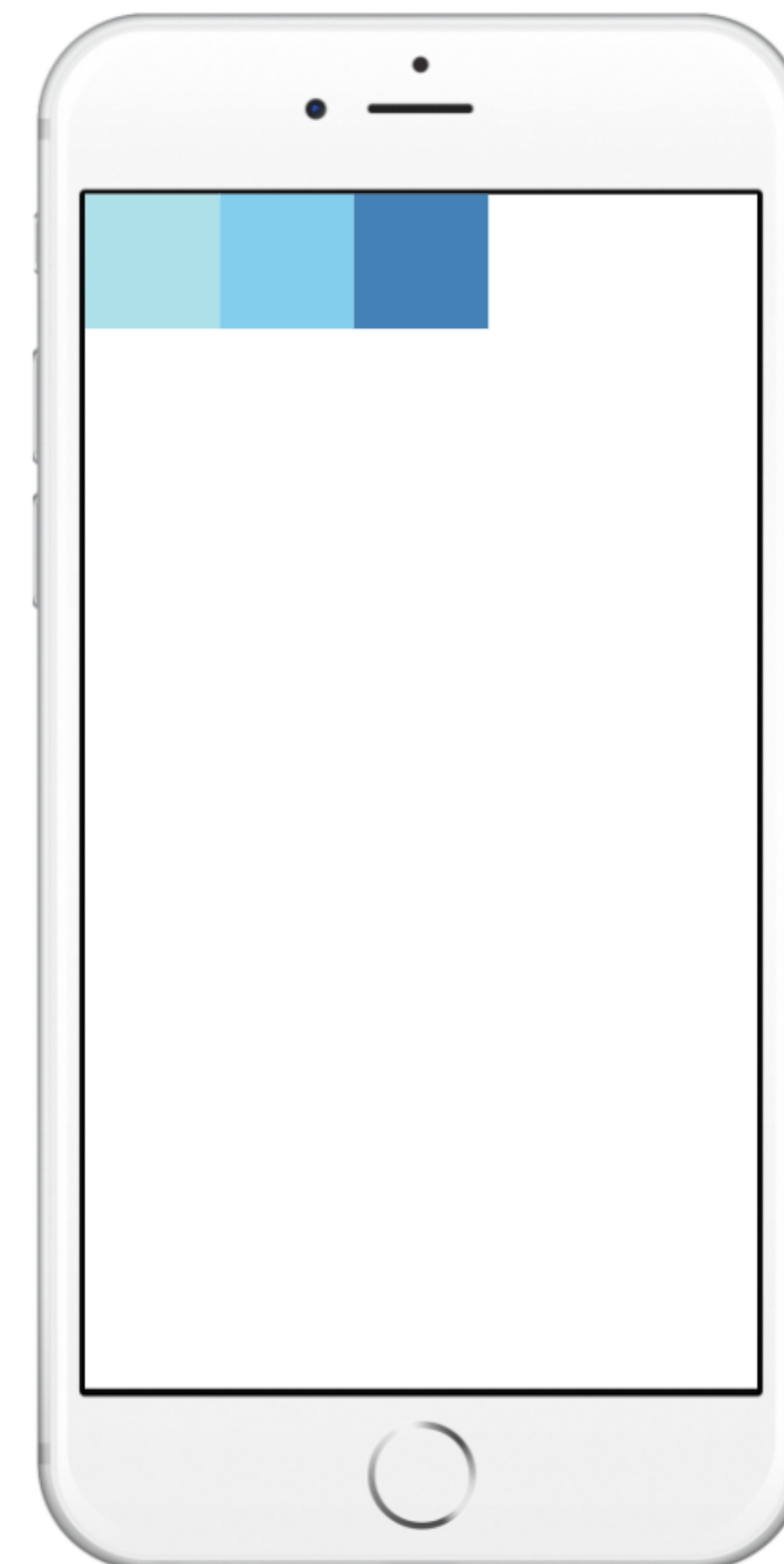
- ◆ Class
- ◆ 箭头函数
- ◆ 解构赋值
- ◆ import / export
- ◆ 参数默认值
- ◆

```
import React, { Component } from 'react'
import {
  View,
  Button
} from 'react-native'

class MyComponent extends Component {
  onBtnPress(msg = 'Hello, aotu guys') {
    alert(msg)
  }
  render() {
    return (
      <View>
        <Button
          title='press me'
          onPress={() => this.onBtnPress()}
        />
      </View>
    )
  }
}
export default MyComponent
```

- FlexBox 布局
- 相对定位/绝对定位
- margin / padding

```
1 import React, { Component } from 'react';
2 import { AppRegistry, View } from 'react-native';
3
4 class FlexDirectionBasics extends Component {
5   render() {
6     return (
7       // Try setting `flexDirection` to `column`.
8       <View style={{flex: 1, flexDirection: 'row'}}>
9         <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />
10        <View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />
11        <View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />
12      </View>
13    );
14  }
15 };
16
17 AppRegistry.registerComponent('AwesomeProject', () => FlexDirectionBasics);
```



TabsView

View 组件 + Flexbox 布局

```
<View style={{flexDirection: 'row'}}>
  <View style={{flex: 1}}><Text>秒杀中</Text></View>
  <View style={{flex: 1}}><Text>秒杀中</Text></View>
  <View style={{flex: 1}}><Text>秒杀中</Text></View>
  <View style={{flex: 1}}><Text>秒杀中</Text></View>
  <View style={{flex: 1}}><Text>秒杀中</Text></View>
</View>
```

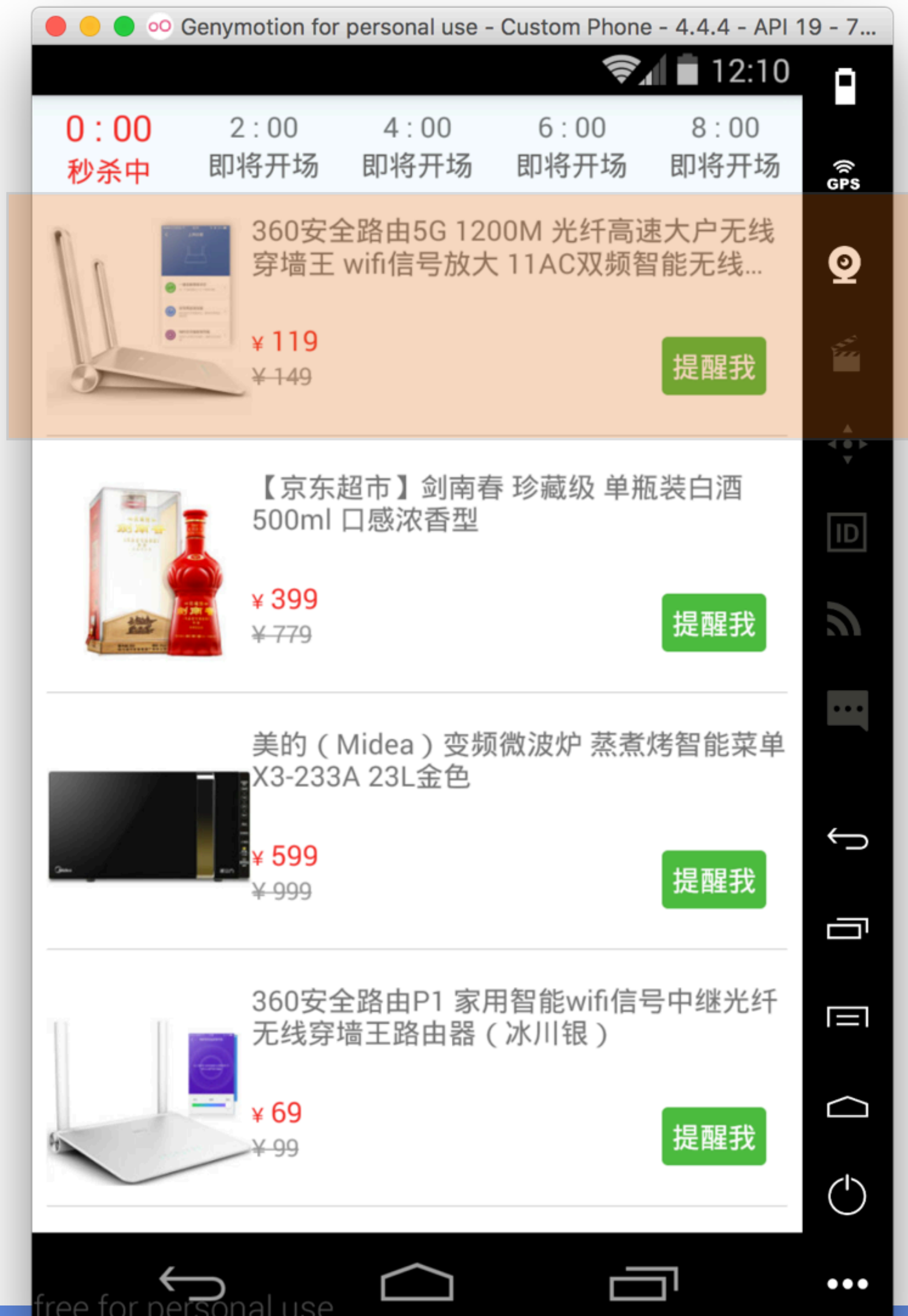


GoodsList

Listview 组件

```
class GoodsList extends Component {
  constructor() {
    this.hasChanged = (r1, r2) => r1 !== r2
    this.ds = new ListView.DataSource(this.hasChanged)
  }

  render() {
    return (
      <ListView
        dataSource={this.ds.cloneWithRows([{}])}
        renderRow={rowData =>
          <GoodsListItem rowData={rowData} />
        }
      />
    )
  }
}
```



GoodsListItem

View + Flexbox + 绝对定位布局

```
<View style={[styles.container, { flexDirection: 'row' }]}>
  <View style={{ width: 120 }}><Image /></View>
  <View style={{ flex: 1 }}>
    <Text>{title}</Text>
    <Text>{price}</Text>
    <Text>{jdPrice}</Text>
    <Text
      style={{
        position: 'absolute',
        bottom: 10,
        right: 10
      }}
    >提醒我</Text>
  </View>
</View>
```

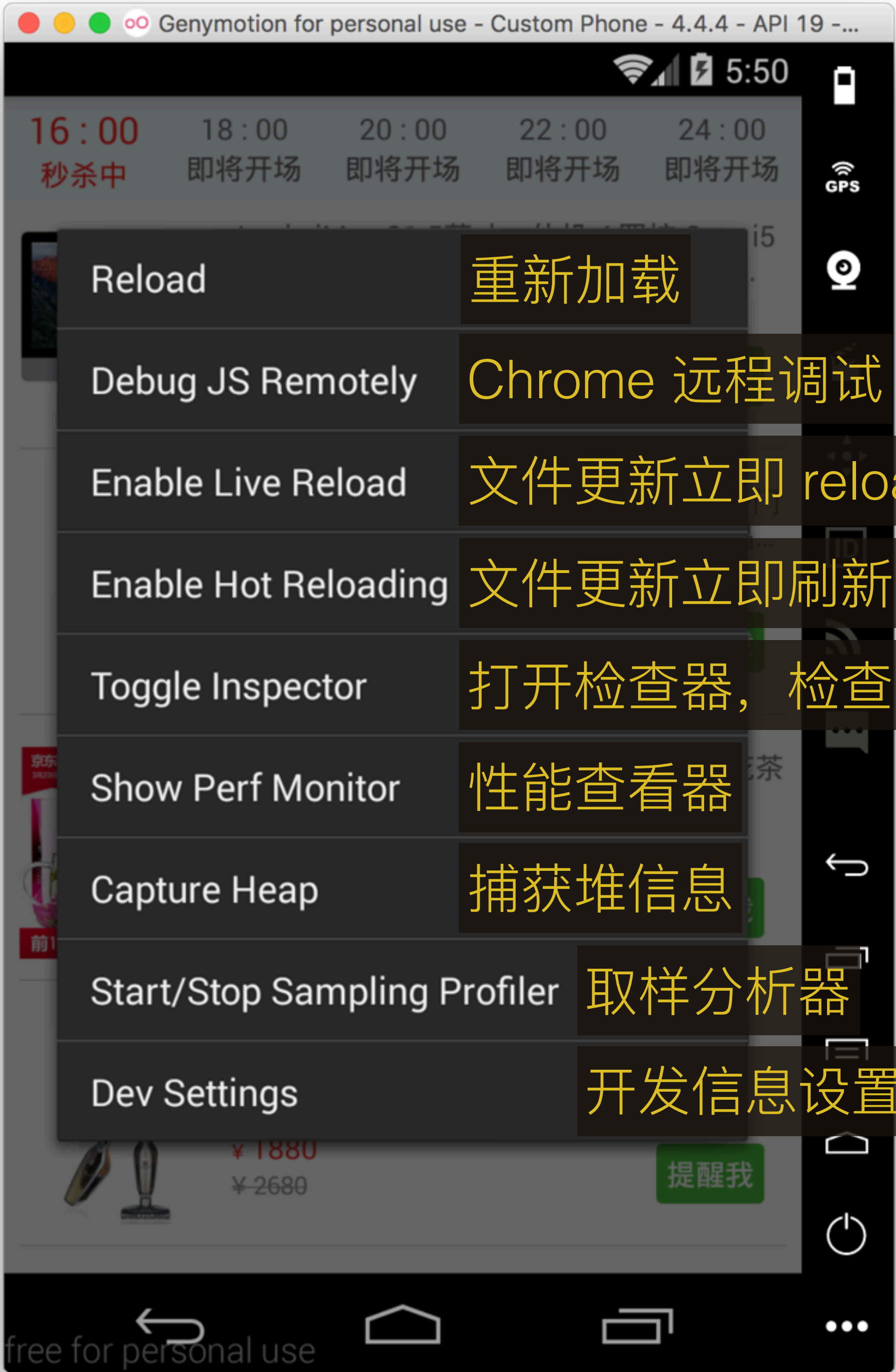
实战代码地址：

<https://github.com/Calvin92/React-Native-Demo>

- ▶ 自带 fetch 异步请求API
- ▶ 与Web一致的API
- ▶ Promise化

- ▶ 自带 fetch 异步请求API
- ▶ 与Web一致的API
- ▶ Promise化

```
fetch(`https://api.m.jd.com/${functionId}`)  
  .then(resp => resp.json())  
  .then(data => {  
    // Do something  
  })  
  .catch(err => {  
    // Handle error  
  })
```



Reload

重新加载

Debug JS Remotely

Chrome 远程调试

Enable Live Reload

文件更新立即 reload, 不保留 state

Enable Hot Reloading

文件更新立即刷新, 保留 state

Toggle Inspector

打开检查器, 检查组件盒模型

Show Perf Monitor

性能查看器

Capture Heap

捕获堆信息

Start/Stop Sampling Profiler

取样分析器

Dev Settings

开发信息设置

调试

02

Dev Settings 中
设置获取 Bundle JS 的地址



3

—

一些坑整理

RN 坑整理地址：

<https://shimo.im/doc/g0XYRhaxvMAHN1j0>

实现图片 placeholder

componentDidMount 中不能操作组件实例

ListView 的 onEndReached 不能触发

ListView 需要在滚动的时候才渲染

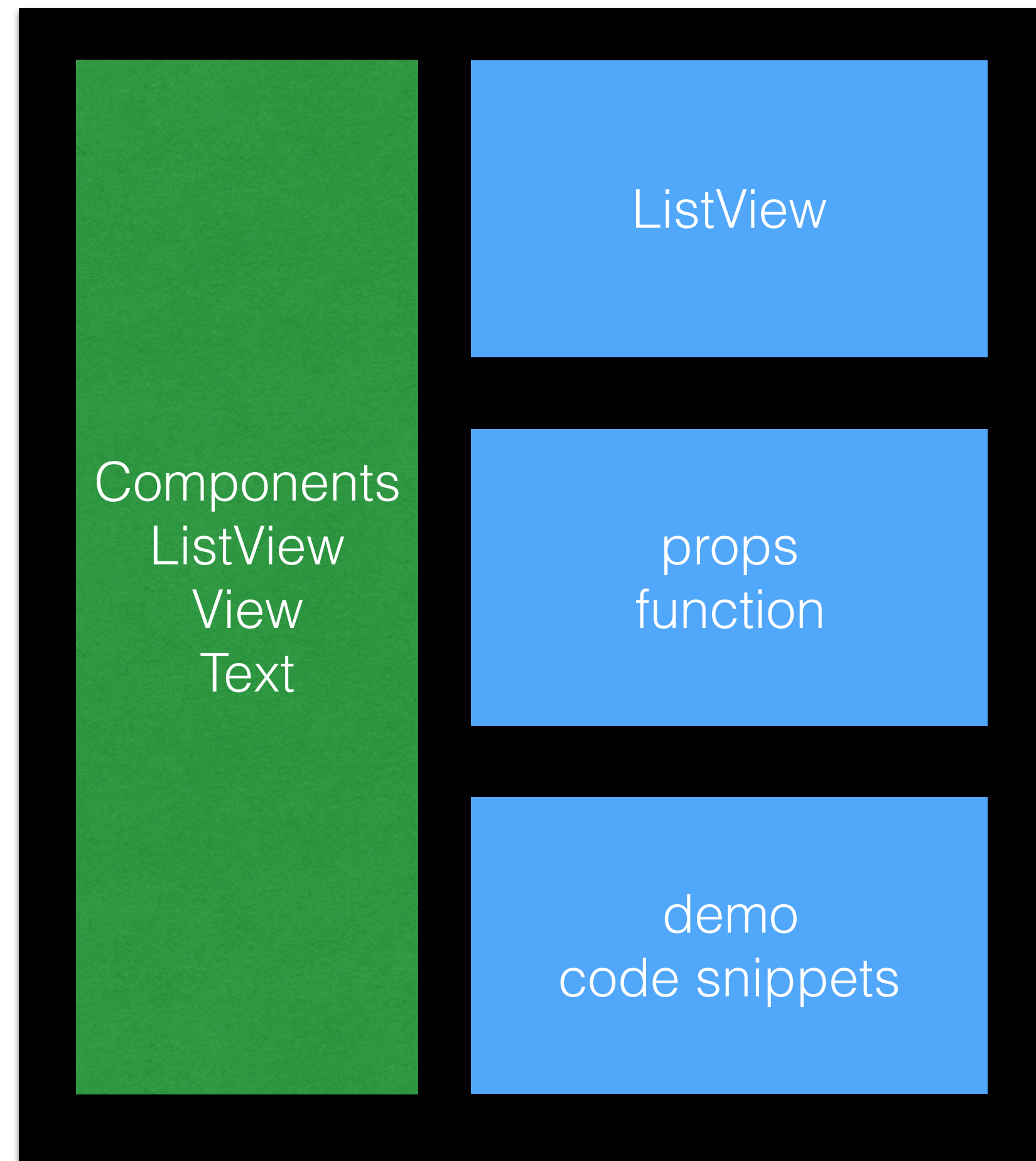
iOS 下 borderRadius 不生效

WebView 在 source 中设置了 baseUrl 后无法渲染

2

项目总结

- ▶ 比原生开发效率更高
- ▶ 一套代码三端跑
- ▶ 学习成本低



RN 官方文档

<http://facebook.github.io/react-native/docs/getting-started.html>

React Native 0.42

Docs

Help

Showcase

Blog

Search docs...

GitHub

React

The Basics

Getting Started

Tutorial

Layout with Flexbox

Handling Text Input

Guides

Handling Touches

Testing

Understanding the CLI

Integration With Existing Apps

Running On Device

Upgrading

Components

ActivityIndicator

Button

DatePickerIOS

DrawerLayoutAndroid

Image

KeyboardAvoidingView

ListView

MapView

Modal

Navigator

NavigatorIOS

Picker

PickerIOS

ProgressBarAndroid

ProgressViewIOS

RefreshControl

ScrollView

SegmentedControlIOS

WebView

WebView renders web content in a native view.

```
import React, { Component } from 'react';
import { WebView } from 'react-native';

class MyWeb extends Component {
  render() {
    return (
      <WebView
        source={{uri: 'https://github.com/facebook/react-native'}}
        style={{marginTop: 20}}
      />
    );
  }
}
```

组件介绍

You can use this component to navigate back and forth in the web view's history and configure various properties for the web content.

Props

View props...

attributes

automaticallyAdjustContentInsets?

属性介绍

Controls whether to adjust the content inset for web views that are placed behind a navigation bar, tab bar, or toolbar. The default value is `true`.

You can edit the content above on GitHub and send us a pull request!

Examples

Edit on GitHub

```
'use strict';

var React = require('react');
var ReactNative = require('react-native');
```

示例

Run this example

课后练习



跟着老司机玩转Node命令行

对于一些没有接触过的东西，或许会因为它的神秘而觉得遥不可及...

#命令行 #NodeJS #老司机



跟着老司机玩转Node命令行

对于一些没有接触过的东西，或许会因为它的神秘而觉得遥不可及...

#命令行 #NodeJS #老司机

```
// 资讯 API 地址 (GET) :  
https://aotu.jd.com/aurora/api/newsView  
?simple=true&channelKey=FRONTEND&page=1&per\_page=3
```

```
// 响应格式 (资讯数组) :  
[  
  {  
    title: '标题',  
    excerpt: '摘要',  
    image: {  
      url: 'http://a.png'  
    },  
    tags: ['命令行', 'NodeJS']  
  }  
]
```

One more thing...

凹凸公开课

<https://aotu.jd.com/share/>

