

本书配套的GitHub文档持续更新中，
保证书中知识在最新版本RN中可用。

Broadview[®]
www.broadview.com.cn

React Native

跨平台移动应用开发 (第2版)

阙喜涛 编著

ES6语法成为主流，鼓励使用Promise机制



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

React Native

跨平台移动应用开发 (第2版)

阙喜涛 编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

React Native 是 Facebook 公司推出的强大的、开源的跨平台移动应用开发框架。它能大幅减少跨平台移动应用开发的工作量（相比原生代码开发能减少至少 50%），并且代码结构清晰、简单易懂。同时 React Native 框架采用模块化的结构，使应用版本的更新迭代非常简单。随着它的日趋成熟，React Native 必然会成为移动应用开发的主流技术。

本书由浅入深，系统地介绍了使用 React Native 框架跨平台开发所需要用到的知识。本书每一章都专注于阐述移动应用开发某一方面的知识，配合若干个原创的、精简的例程，能让读者系统、快速地掌握该方面知识。

本书的结构是按一个有基本的编程基础知识，刚学习 JavaScript 基本语法的读者的最佳学习路径来编写的。有一定基础的开发人员也可以将本书作为一本 React Native 开发的“字典”来使用，在开发时可以快速查找相关知识点的细节。

希望通过本书，能将最新的 React Native 开发技术介绍给国内广大开发者，让国内移动应用开发进入一个新的时代，让开发者用比较少的精力就能高效、美观地完成移动应用开发。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

React Native 跨平台移动应用开发 / 阙喜涛编著. —2 版. —北京：电子工业出版社，2017.5
ISBN 978-7-121-31273-1

I. ①R… II. ①阙… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字（2017）第 069667 号

策划编辑：孙学瑛

责任编辑：石 倩

印 刷：三河市华成印务有限公司

装 订：三河市华成印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱

邮编：100036

开 本：787×1092 1/16 印张：26

字数：678 千字

版 次：2016 年 6 月第 1 版

2017 年 5 月第 2 版

印 次：2017 年 5 月第 1 次印刷

定 价：99.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819 faq@phei.com.cn。

前言

React Native 开发的发展

在本书第 1 版的前言中，笔者写下了一个预言：我看到了一个新的伏地魔正在崛起，他会统治这个世界，他那高高的巫师帽上，React Native 这个名字不时地散发着邪恶的光芒！

距离本书第 1 版上市将近一年了。在这一年中，React Native 发展得如何呢？很明显，目前 React Native 还没有统治这个世界，但一大批知名手机应用放弃了老版本使用原生语言开发的做法而在新版本开发中使用了 **React Native** 跨平台开发技术（未特殊注明均是用 RN 实现了跨平台应用）。国内有：京东、手机百度、天猫（iPad 版）、携程旅行、蜂鸟众包（饿了么配送端移动应用）、腾讯课堂、QQ、QQ 音乐与 QQ 空间（这三个均为 Android 版）等；国外有：Facebook 系列 APP、Instagram、Airbnb、沃尔玛（Walmart）、彭博（Bloomberg）等。在服务器侧，微软的免费开源平台 CodePush 支持了 React Native 移动应用，Google 统计平台支持了 React Native 移动应用，极光推送平台支持了 React Native 移动应用……越来越多的开发者与开发团队意识到了 React Native 开发技术的优越性，加入到了 React Native 开发社区中。

React Native 自身也从一个高速发展的阶段进入到一个稍平稳些发展的阶段。从 RN 面世开始，它以每半个月升级一次小版本的速度从 0.1.0 升级到了 0.40.0。但在 2017 年 2 月，RN 社区决定将升级周期延长到 1 个月一个小版本升级。这个决定从某方面反映出目前 RN 已经足够完善。

这一年中，除了努力完善自身，RN 在两个方向上的发力让 RN 变得对开发者更友好。一是抛弃老旧的 ES5 语法，大步进入 ES6 语法开发时代。二是全力抛弃让开发者难受的回调函数机制，代之以结构清晰、易读易书写的 Promise 机制。

以上两点归结到一个目标就是让开发更简单、更舒服。本来一个框架实现跨平台开发就已经让开发者很舒服，现在变得更简单、更舒服，我想我不用再解释为什么这么多开发者与开发团队选择 React Native 了吧。

写作本书的目的

React Native 项目代码是开源的，项目代码中的例程也是开源的，项目文档也是在网上公开的，

所有人都可以非常方便地获取浏览。但目前 React Native 的文档假设它的读者有相当深厚的开发方面的知识 & 经验，并且有 React.js 开发基础。也就是说，React Native 的公开文档对初学者并不是非常友好的。React Native 项目中的例程代码对初学者而言同样有些高深。在各个 React Native 论坛上，React Native 的文档不够好是大家的共识。

笔者非常努力地把这本书写好，让它由简入深、通俗易懂。本书绝对不是粗制滥造、简单翻译的书籍。书中讲述的知识点结构、例程设计都倾注了笔者大量的心血，与网上公开文档的差别非常大。本书对读者最大的价值是：把读者通过阅读网上公开文档、项目例程学习 RN 开发技术所需要的 800 小时左右的时间（视个人基础有所不同）缩短为 300 小时左右。

本书中的 6.5 节、第 11 章、14.5 节、第 16 章等都是笔者精心整理调研出来的技术，在官方文档中没有提及，希望能让更多的开发者享受到 React Native 开发的便利性。

希望通过本书，能将 React Native 开发技术介绍给国内广大开发者，让国内移动应用开发进入一个新的时代，让开发者用比较少的精力就能高效、美观地完成移动应用开发。

读者对象

本书的读者对象如下：

- Android 平台移动应用研发工程师
- iOS 平台移动应用研发工程师
- JavaScript 软件研发工程师
- 计算机相关专业的学生

如何阅读本书

React、React.js、React Native

对于初学者常见的困惑是弄不清 React、React.js、React Native 三者之间的关系。这是情有可原的。初学者经常发现在很多讨论 React Native 技术的资料中，突然会说到 React.js，过一会儿又好像说到 React 基础框架。因此在讨论如何阅读本书之前，有必要先说明一下这三者的关系。React 是基础框架，是一套基础设计实现理念，开发者不能直接使用它来开发移动应用或者网页。在它之上发展出了 React.js 框架用来开发网页，发展出来 React Native 用来开发移动应用。因为 React 基础框架与 React.js 框架是同时出现、同时进化发展的，这就造成了 React 基础框架的基本概念、设计思想都是在 React.js 的相关文档中描述的。后来，Facebook 推出 React Native 后，也没有把 React 的相关概念文档从 React.js 文档中分离出来，这就导致了学 React Native 要去看 React.js 文档，说 React Native 不时会说到 React.js 的情况。如果开发者只想关注移动应用开发，那么在 React.js 的如何开发网页的文档中整理出 React 基础框架知识不是一个轻松的活儿。现在，本书推出了，有了这本书，读者可以不用再去看 React.js 文档了。

预备知识

学习 React Native 开发需要基础的 JavaScript 编程知识。我估计有一部分读者可以在某个手机平台使用该平台原生语言进行移动应用开发，但对 JavaScript 只是有所耳闻。如果是这种情况，不用担心。读者只需要随便找一本 JavaScript 入门级的书籍（甚至是网上的教学性网页），阅读关于基础语法的章节，花上一天时间（包括找资料的时间）学习相关知识点，接下来就可以通过本书学习 React Native 开发了。所需要的知识点有：

（1）JavaScript 语法。包括语句、注释、变量、数据类型、数组（注意关联数组，Java、C++ 的数组中没有这个概念）、对象的基本知识。

（2）操作运算符。这个与 Java、C++ 基本上是一样的，读者快速过一下就行。

（3）条件语句、循环语句、switch 语句。这与 Java、C++ 基本上是一样的，读者快速过一下就行了。

（4）函数。JavaScript 中函数也是一种变量，知道了这一点，其他与 Java、C++ 基本上是一样的，读者快速过一下就行了。

（5）对象。JavaScript 的对象定义、实现比 Java、C++ 宽松很多。读者得稍微适应一下。

这些基本的知识点，大多数与 Java 或者 Objective-C 的相关知识点很类似。如果读者有基础，阅读这些知识点最多只需要一天的时间。读者不需要搭建 JavaScript 的开发环境来练习、巩固这些知识点，因为在 React Native 的开发环境中编写代码就可以练习这些基本的 JavaScript 知识点，在 React Native 学习中就会巩固这些 JavaScript 基本知识。

React Native 在开发中用到了其他 JavaScript 的高级知识点与 ES 6 的一些新特性，但读者不需要马上去学习这些内容。在通过本书学习 React Native 开发的过程中若需要使用 JavaScript 高级知识点，会指出在附录 A 的什么位置讲解了这些高级知识点，便于读者快速查看。附录 A 不是 JavaScript 知识点的全面讲解，只是让读者对 React Native 开发中需要使用到的 JavaScript 知识点有足够理解以进行 React Native 开发。

相关下载地址

笔者希望读者在阅读本书时，能在理解的基础上将例程代码输入到电脑中。输入的过程是一个消化吸收的过程。输入完成后，运行代码，并且按照提示或者针对自己有疑虑的地方进行修改，以便深入理解各个知识点。

正因为如此，本书前面章节中那些短小精悍的例程并没有附在一张光盘上，或者在网上提供下载地址，而是需要读者自己手动输入电脑。

本书还有一些不需要读者手动输入的代码，笔者在 GitHub 上提供了一个网址供读者自行下载。网址是：<https://github.com/es6rn>。

本书结构

本书讨论的 React Native 开发特性覆盖了 2017 年 3 月 2 日发布的 React Native 0.42.0 版本的绝大部分特性。没有讨论的部分在书末有提及。

首先需要说明的是，本书讨论的知识以跨平台（Android 平台与 iOS 平台）开发为主，书中各章节的绝大部分知识点都是跨平台实现的。只有极小部分是分平台实现的，这一小部分在讨论前都会说明该部分知识适用于哪个平台。

本书的结构是按一个有基本的编程基础知识，刚学习 JavaScript 基本语法的读者的最佳学习路径来编写的。通过一个个精简的例程，清晰地阐述一个个基本思想。例程尽可能地精简，并且所有例程都是笔者花了大量时间为初学者构思而成的。

本书体例说明

代码与代码说明

本书例程中有大量的代码说明，通过注释的方式与代码同时展示出来。例如：

```
var IncomingCall = React.createClass({
  watcher: null,           //用来记录监视器
  startFromLeft:true,     //用来判断用户最先按下的是最左侧还是最右侧
  moveNeedhandle:false,   //用来判断监测到的移动事件是否需要处理
});
```

注释以粗体字显示以提醒读者注意。读者在自己的开发环境中输入例程时不需要输入注释。

React Native 代码中的 JSX 部分代码不能使用这种注释方式，本书为了统一注释风格还是使用了这种注释方式。读者明白了这一点后，在自己输入代码试验时，请将代码中的注释自行去掉。

注意和提示

注意和提示，是需要提醒读者特别注意的内容。在本书中使用带背景色的字体显示。

致谢

感谢我的父亲阙光金老师与我的母亲袁雪英老师从小到大给我的无私的爱。很抱歉无论我如何努力，也无法回报二老深恩的万分之一。

感谢我的姐姐阙喜戎与姐夫王纯，没有你们的鼎力支持，就没有今天的我。

感谢王汝馨伯父与曾钰伯母，谢谢你们对我的关怀与照顾。

感谢廖建新教授、饶牧老师在我学习工作期间对我的关怀与指导。感谢在我 7 年北京邮电大学学习期间为我授业解惑的所有老师，谢谢你们！

感谢 React Native 开发团队，感谢所有参与 React Native 开发的贡献者。无数移动开发者因为

你们无私的奉献而受益。

感谢电子工业出版社郭立总经理、孙学瑛编辑等为审校此书而付出的辛勤工作，以及为此书能快速出版而付出的巨大努力。你们辛苦了！

感谢在工作和生活中帮助过我的所有人，感谢你们，正是因为有了你们，才有了本书的面世。

关于勘误

虽然花了很多时间和精力去核对书中的文字、代码和图片，但因为时间仓促和水平有限，书中仍难免会有一些错误和纰漏，如果大家发现什么问题，请反馈给我，相关信息可在下载本书代码的 GitHub 页面反馈。

轻松注册成为博文视点社区用户（www.broadview.com.cn），您即可享受以下服务：

- 提交勘误：您对书中内容的修改意见可在【提交勘误】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- 与作者交流：在页面下方【读者评论】处留下您的疑问或观点，与作者和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/31273>

二维码：



目 录

第 1 章	React Native	1
1.1	React Native 开发特点	2
1.1.1	一次学习，随处编写	2
1.1.2	混合开发	2
1.1.3	高效的移动应用开发	3
1.1.4	高效的移动应用开发调试	4
1.1.5	灵活高效的应用热更新	4
1.1.6	有效降低移动应用安装包体积	5
1.1.7	学习门槛低、开发难度低	5
1.1.8	开发软硬件要求低	6
1.1.9	使用 React Native 开发的代价	6
1.2	React Native 开发环境搭建	7
1.2.1	开发环境搭建起点	8
1.2.2	Windows 操作系统下 React Native 开发环境搭建	8
1.2.3	苹果操作系统下 React Native 开发环境搭建	10
1.2.4	查看与删除使用 npm 命令安装的软件	11
1.3	代码编辑环境搭建	11
1.3.1	Visual Studio Code	11
1.3.2	Sublime Text 3	11
1.3.3	WebStorm	13
1.4	键盘使用习惯	13
1.5	React Dev Tools 安装	14
第 2 章	状态机思维与状态机变量	17
2.1	初始化项目	17
2.1.1	初始化 RN 最新版项目	17
2.1.2	初始化 RN 旧版本项目	18
2.2	运行项目	19
2.2.1	使用 Android 手机调测准备事项	19
2.2.2	第一次运行 Android 平台 RN 项目	19
2.2.3	再次在 Android 平台运行 RN 项目	21

2.2.4	使用 iPhone 设备调试注意事项	22
2.2.5	首次在 iOS 平台运行 RN 项目	23
2.2.6	再次在 iOS 平台运行 RN 项目	27
2.2.7	Sublime Text 3 对 JSX 语法的支持	27
2.2.8	修改 JSX 代码	28
2.2.9	启动调试工具	30
2.2.10	使用 ADM 工具调试	35
2.3	构建登录页面	37
2.4	React Native 代码执行逻辑	41
2.5	UI 框架工作基本机制	41
2.5.1	状态机思维	42
2.5.2	“冒充常量”的状态机变量	44
2.5.3	“无处安放”的状态机变量	45
2.5.4	“努力瘦身”的状态机变量	46
2.6	React Native 组件间通信	46
2.7	深入理解 UI 重新渲染的过程	47
2.7.1	合并状态机变量	47
2.7.2	判断是否渲染	49
2.7.3	强制启动渲染	50
2.7.4	渲染过程	50
2.7.5	合并状态机变量的最简语法	50
2.8	React Native 组件的成员变量	51
2.9	React Native 组件的静态变量、静态函数	52
2.10	组件回调函数的绑定	52
第 3 章	页面导航、弹出框及深入理解属性	55
3.1	分离注册组件、组件平台自适应	55
3.1.1	分离注册组件	55
3.1.2	组件平台自适应	55
3.1.3	平台检测	56
3.2	导航组件、挂接注册组件	56
3.3	挂接注册等待组件	58
3.4	Navigator 组件	61
3.4.1	push 与 pop	61
3.4.2	replace 函数	62
3.5	React Native 中颜色类型的值	62
3.6	BackAndroid API	63
3.7	属性确认	63
3.8	指定属性默认值	66
3.9	Alert 应用程序编程接口	66

3.9.1 弹出确认框.....	66
3.9.2 弹出选择框.....	67
3.10 带导航栏的页面导航.....	69
第4章 混合开发基础篇.....	70
4.1 iOS 平台混合开发.....	70
4.1.1 与 iOS 侧原生代码消息互通.....	71
4.1.2 React Native 代码到 iOS 原生代码的消息.....	72
4.1.3 iOS 原生代码到 React Native 代码的消息.....	75
4.1.4 与 iOS OC 原生代码界面切换.....	77
4.1.5 应用初始界面设定.....	77
4.1.6 iOS 混合开发中传递的参数类型.....	78
4.1.7 混合开发中的多线程使用.....	78
4.1.8 原生代码实现 Promise 机制.....	79
4.1.9 跨语言常量.....	82
4.2 Android 平台混合开发.....	82
4.2.1 与 Android 原生代码消息互通.....	83
4.2.2 React Native 代码到 Android 原生代码的消息.....	84
4.2.3 与 Android 原生代码界面的切换.....	87
4.2.4 Android 原生代码到 React Native 代码的消息.....	91
4.2.5 应用初始界面设定.....	94
4.2.6 传递的参数类型.....	94
4.2.7 回调函数与 Promise 机制.....	95
4.2.8 监听 ActivityResult 与 Android 生命周期事件.....	97
4.2.9 混合开发中的多线程机制.....	98
4.2.10 跨语言常量.....	98
第5章 flexbox 布局、View、Image 与可触摸组件.....	100
5.1 flexbox 布局.....	100
5.1.1 位置及宽、高相关样式键.....	102
5.1.2 决定子组件排列规则的键.....	102
5.1.3 决定组件显示规则的键.....	106
5.1.4 边框、空隙与填充.....	109
5.1.5 zIndex.....	110
5.1.6 组件多样式声明与动态样式声明.....	110
5.2 View 组件.....	111
5.2.1 View 组件的颜色与边框.....	111
5.2.2 View 组件的阴影与其他视觉效果.....	113
5.2.3 View 组件的变形.....	114
5.2.4 View 组件的回调函数.....	117

5.2.5	View 组件的其他属性	119
5.2.6	设备放置状态、根 View 与 onLayout 回调函数	120
5.2.7	pointerEvents 属性	123
5.3	Image 组件	125
5.3.1	加载网络图片	125
5.3.2	加载静态图片资源	127
5.3.3	加载资源文件中的图片	128
5.3.4	动态加载手机中的图片资源	129
5.3.5	Image 组件的样式	129
5.3.6	Image 组件显示特性	131
5.3.7	Image 组件的其他属性	134
5.3.8	Image 组件的缓存	134
5.3.9	尽量使用网络图片	135
5.4	可触摸组件	135
5.4.1	可触摸组件类型	135
5.4.2	TouchableOpacity 组件	136
5.4.3	TouchableHighlight 组件	137
5.4.4	其他属性	138
5.5	使用导航栏的导航框架	139
5.5.1	导航栏自定义组件的实现	140
5.5.2	调用自定义组件	142
第 6 章	Text、TextInput 等相关知识	145
6.1	Text 组件	145
6.1.1	样式键设置	145
6.1.2	其他属性	147
6.1.3	Text 组件的嵌套	149
6.1.4	文本显示的阴影效果	150
6.1.5	Text 居中显示	151
6.1.6	在字符串中插入图像	153
6.2	Text 组件在两个平台上的不同表现	155
6.2.1	只指定 fontSize, 不指定 height	155
6.2.2	只指定 height, 不指定 fontSize	156
6.2.3	fontSize 等于 height	156
6.2.4	height 大于 fontSize	156
6.2.5	边框在两个平台上的不同表现	157
6.3	TextInput 组件	158
6.3.1	TextInput 组件样式键	158
6.3.2	TextInput 组件的属性	158
6.3.3	TextInput 组件 iOS 平台专有属性	160

6.3.4	TextInput 组件 Android 平台专有属性	161
6.3.5	TextInput 组件的成员函数	161
6.4	TextInput 组件在两个平台上的不同表现	161
6.4.1	Android 平台的输入下划线	162
6.4.2	父组件的 alignItems 键失效	162
6.4.3	只指定 fontSize，不指定 height	164
6.4.4	height 等于 fontSize	164
6.4.5	height 大于 fontSize	164
6.4.6	边框在两个平台上的不同表现	165
6.5	TextInput 组件的生命周期	165
6.5.1	获得焦点	165
6.5.2	用户输入	166
6.5.3	用户按下提交键	166
6.5.4	失去焦点	166
6.5.5	生命周期中的其他事件	167
6.6	Keyboard API、软键盘与键盘事件	168
6.6.1	Keyboard API	168
6.6.2	键盘事件处理例程	168
6.7	组件的引用	171
6.7.1	定义组件引用	171
6.7.2	得到系统定义的组件引用	171
6.7.3	调用组件的公开成员函数	172
6.7.4	重新设定组件的属性	172
6.7.5	获得组件的位置	174
6.8	跨平台状态栏组件	175
6.8.1	StatusBar 组件属性	175
6.8.2	StatusBar 组件使用示例	176
6.8.3	手机状态栏在开发中的处理	176
6.9	高度自增长的扩展 TextInput 组件	177
6.10	访问操作系统剪贴板	179
第 7 章	组件生命周期、数据存储及 React Native 应用实现步骤	182
7.1	组件生命周期	182
7.1.1	constructor	182
7.1.2	componentWillMount	182
7.1.3	componentDidMount	183
7.1.4	componentWillReceiveProps	183
7.1.5	shouldComponentUpdate	183
7.1.6	componentWillUpdate	184
7.1.7	componentDidUpdate	184

7.1.8	componentWillUnmount	184
7.2	读取 JSON 文件	185
7.3	数据持久化操作	186
7.3.1	写入数据、错误捕捉	186
7.3.2	读取数据	188
7.3.3	AsyncStorage API 存储数据的无序性	190
7.3.4	删除数据	191
7.3.5	修改数据	192
7.3.6	JSON 对象存储	192
7.3.7	读取 JSON 对象	193
7.4	数据表操作	193
7.5	React Native 应用实现步骤、日记例程（上）	193
7.5.1	应用原型	194
7.5.2	基础组件结构设计	195
7.5.3	使用 React Native 组件搭建静态界面	197
7.5.4	React Native 组件分层	204
7.5.5	实现各组件业务逻辑	206
7.5.6	日记例程（上）总结	206
第 8 章	ScrollView 和 ListView	218
8.1	ScrollView 组件	218
8.1.1	ScrollView 组件属性	218
8.1.2	ScrollView 组件 iOS 平台专有属性	219
8.1.3	ScrollView 组件 Android 平台专有属性	220
8.1.4	ScrollView 组件的公开成员函数	221
8.1.5	RefreshControl 组件	221
8.1.6	ScrollView 组件基本用法	222
8.2	ListView 组件	224
8.2.1	ListView 组件的回调函数	225
8.2.2	ListView 组件的其他属性	226
8.2.3	ListView 组件的成员函数	226
8.3	简单的列表	226
8.3.1	准备列表的数据源	227
8.3.2	声明状态机变量	227
8.3.3	将数据源中的数据复制到 DataSource 中	228
8.3.4	定义如何渲染列表中的每一行	228
8.3.5	实现简单的列表	229
8.3.6	列表栏的高级处理	237
8.4	带分段标志的列表	237
8.4.1	准备数据源	238

8.4.2	声明状态机变量	238
8.4.3	将数据源中的数据复制到 DataSource 中	239
8.4.4	定义如何渲染每个分栏	240
8.4.5	定义如何渲染首、尾栏	240
8.4.6	列表间隔渲染	240
8.4.7	实现带分段标志的列表	241
8.4.8	总结	241
8.5	日记例程（下）总结	241
第 9 章	等待提示、进度条和 Switch	242
9.1	ActivityIndicator 组件	242
9.1.1	ActivityIndicator 组件样式设置	242
9.1.2	ActivityIndicator 其他属性	242
9.1.3	等待提示例程	242
9.2	iOS 进度条组件	243
9.2.1	React Native 框架中定时器的使用	243
9.2.2	ProgressViewIOS 组件样式设置	244
9.2.3	ProgressViewIOS 其他属性	244
9.2.4	iOS 平台进度条	245
9.3	Android 平台进度条组件	246
9.4	Switch 组件	246
9.4.1	Switch 组件样式设置	246
9.4.2	Switch 其他属性	246
9.4.3	Switch 组件的使用	247
第 10 章	导航组件与 Modal	249
10.1	导航组件的属性	249
10.1.1	回调函数	249
10.1.2	其他属性	250
10.2	导航器	250
10.3	NavigationBar	251
10.4	导航例程	252
10.5	Modal 组件	257
10.6	Modal 组件与 Navigator 组件的配合	258
10.7	Modal 组件例程	258
10.7.1	实现自定义 Modal 组件	258
10.7.2	使用自定义 Modal 组件	260
10.7.3	Modal 组件与 Alert 组件	262
10.7.4	总结	262

第 11 章 手势识别	263
11.1 PanResponder API	263
11.2 监视器	263
11.2.1 指定监视区域	264
11.2.2 定义监视器相关变量	264
11.2.3 准备监视器的事件处理函数	264
11.2.4 建立监视器	265
11.2.5 将监视器与监视区域挂接	265
11.3 监视事件的生命周期	265
11.3.1 单次点击事件的生命周期	266
11.3.2 单次点击事件处理	268
11.3.3 移动手势事件的生命周期	268
11.3.4 监视器异常事件	270
11.4 手势识别处理例程	270
11.4.1 单点手势——点击、拖动选择百分比参数	270
11.4.2 单点手势——带导槽的滑动来电接听或拒接界面	273
11.4.3 单点手势——滑动解锁屏幕界面	275
11.4.4 单点手势——单点任意方向拉动选择界面	277
11.4.5 两点手势	281
第 12 章 网络	282
12.1 获取网络状态	282
12.1.1 得到当前网络状态	282
12.1.2 监听网络状态改变事件	283
12.1.3 简单判断是否有网络连接	284
12.1.4 判断当前连接是否收费	284
12.2 搭建调试用 HTTP、HTTPS 服务器	284
12.2.1 搭建 HTTP 测试服务器	285
12.2.2 搭建 HTTPS 测试服务器	285
12.3 通过 HTTP、HTTPS 与网络侧交换数据	286
12.3.1 发送请求	286
12.3.2 接收响应	288
12.3.3 HTTP 通信完整例程	290
12.3.4 使用 HTTP 消息向服务器上传手机中的照片	291
12.4 在 React Native 开发中使用 AJAX 技术	291
12.5 WebSocket	292
12.5.1 WebSocket 服务器的创建	292
12.5.2 在 RN 中使用 WebSocket 通信	293
12.6 推送	294
12.6.1 在 Android 平台上使用推送技术	294

- 12.6.2 在 iOS 平台上使用推送技术..... 295
- 12.7 Linking API..... 296
 - 12.7.1 处理发给本应用的链接..... 296
 - 12.7.2 打开外部链接..... 297
- 第 13 章 网页浏览器、音视频媒体播放..... 299
 - 13.1 WebView 组件样式设置..... 299
 - 13.2 WebView 组件其他属性..... 299
 - 13.2.1 非回调函数属性..... 299
 - 13.2.2 回调函数属性..... 300
 - 13.2.3 平台独有属性..... 300
 - 13.2.4 WebView 组件成员函数..... 301
 - 13.3 网页浏览器使用例程..... 301
 - 13.3.1 浏览网页例程..... 301
 - 13.3.2 加载本地网页例程..... 304
 - 13.4 音视频媒体播放..... 306
- 第 14 章 应用权限与图库操作..... 307
 - 14.1 应用权限..... 307
 - 14.1.1 iOS 平台应用权限..... 307
 - 14.1.2 Android 平台应用权限..... 307
 - 14.2 React Native 开发中 iOS 平台链接库的使用..... 310
 - 14.3 获取手机中所有的图片信息..... 312
 - 14.4 图片信息详解..... 314
 - 14.4.1 Android 平台图片信息..... 314
 - 14.4.2 iOS 平台图片信息..... 315
 - 14.5 显示从 CameraRoll API 得到的图片..... 315
 - 14.6 为用户提供图片选择界面..... 316
 - 14.7 图片的保存与读取显示..... 318
 - 14.7.1 保存图片数据..... 319
 - 14.7.2 读取并显示图片..... 319
 - 14.8 将图片通过 POST 消息上传..... 320
 - 14.8.1 搭建配合测试服务器..... 320
 - 14.8.2 上传图片例程..... 320
 - 14.9 裁剪图片..... 323
 - 14.9.1 裁剪图片..... 323
 - 14.9.2 使用裁剪后的图片..... 324
- 第 15 章 选择器、位置相关和应用状态..... 325
 - 15.1 日期、时间选择器..... 325
 - 15.1.1 DatePickerAndroid API..... 325