



React Native

跨平台移动应用开发

阙喜涛 编著

React Native 跨平台移动应用开发

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 公司最新推出的, 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, 并且代码结构清晰、简单易懂。同时 React Native 框架采用模块化结构, 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某一方面的知识, 配合若干个原创的、精小的例程, 能让读者系统、快速地掌握该方面知识。

本书是按照有基本的编程基础知识, 刚学习 JavaScript 基本语法的读者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 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. —北京: 电子工业出版社, 2016.6
ISBN 978-7-121-28707-7

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-(2016) 第 094122 号

策划编辑: 孙学瑛

责任编辑: 葛 娜

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: 22.75

字数: 596 千字

版 次: 2016 年 6 月第 1 版

印 次: 2016 年 6 月第 1 次印刷

定 价: 79.00 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 zlt@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

我对移动应用开发大势的粗浅理解

我必须向大家坦白：我现在是个没有良心的人。因为写这本书不仅把我的良心用完了，还透支了很多。本书所有的示例代码都是原创，并且在代码旁有注释解说，绝没有扔一大段代码给读者自己慢慢看的情况。

仔细想来，所有主流手机平台应用开发我都涉足过，它们是 J2ME、Symbian、Series 60 开发平台、Windows Mobile、Android、iOS、Windows Phone。许多年前我写过一本 Series 60 平台移动应用开发的书。当我写到一大半时，Symbian 手机操作系统、Nokia 手机、Series 60 开发平台这片森林从地球表面消失了……

如果问这么多年我对移动应用开发最深的感触是什么，那就是移动应用开发太特么的难了！难就难在应用需要能运行在不同的手机上。开发者的代码要适配不同的手机操作系统（这意味着要使用不同的编程语言编写）、不同的手机硬件能力（比如开发者的应用需要使用 GPS 定位，然后发现某型号手机没有 GPS 定位功能）、不同的手机屏幕分辨率（想象一下当开发者发现应用程序在另一种屏幕分辨率下显示得乱七八糟时的惊喜）。

随着科技的发展，手机开发也在向好的方向不停地转变。Android 与 iOS 两大手机操作系统称霸江湖（这种稳定对开发者是一件好事儿，相互竞争也让两大操作系统都在不断地提升完善）。手机硬件配置越来越强大，能力越来越强大，价格越来越平易近人。手机现在已经成为了普通老百姓无时无刻不随身携带的电脑。伴随着这种趋势，市场对移动应用开发的需求也越来越多，并且要求越来越高。

这么多年来，移动应用开发者做梦都希望有一个能跨平台的开发工具，让他们不要把同一个移动应用使用不同的开发语言写两遍（或者三四遍）。但跨平台开发工具的实现很难，很多先驱者倒在了这条路上。直到 Facebook 给大家带来了 React Native。

让我以一个预言结束我的唠叨吧。在西方传说故事里那些没良心的巫师的预言忒准了，所以请读者对我的预言稍微有些信心。嗯，我的水晶球呢？啊，在这里。（装模作样好长一段时间）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！

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，让它从简入深，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，怎么突然说到 React.js 上去了？过一会儿怎么又好像说到 React 基础框架上去了？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。这就导致出现了学 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来 React 基础框架知识是个不轻松的活儿。现在，本书推出了，有了这本书，读者可以不用再去查看 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了。

（3）条件语句、循环语句、switch 语句。这个与 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（半天看完也不是难事儿）。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 开发过程中需要使用到的 JavaScript 高级知识点时，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 知识点的理解足够进行 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/xitaoque>。

本书结构

本书讨论的 React Native 开发特性覆盖了 2016 年 3 月 2 日发布的 React Native 0.21.0 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 基本语法的读者最佳学习路径来编写的。通过一个个精简的例程，阐述清楚一个个基本思想。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
```

注释以黑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，在自己输入代码实践时，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。感谢在我七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、孙学瑛编辑等审校此书的辛勤工作，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。

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 UI 开发.....	3
1.1.4	高效的 UI 调试.....	4
1.1.5	学习门槛低、开发难度低.....	4
1.1.6	开发软硬件要求低.....	5
1.1.7	使用 React Native 开发的代价.....	5
1.1.8	为什么 React Native 尚未流行.....	7
1.2	React Native 开发环境搭建.....	7
1.2.1	开发环境搭建起点.....	7
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.....	9
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	Sublime Text 3.....	11
1.3.2	开发用插件.....	11
1.3.3	Sublime 界面风格选择.....	13
1.3.4	键盘使用习惯.....	13
1.4	React Native Dev tool 安装.....	14
第 2 章	状态机思维与状态机变量.....	16
2.1	初始化项目.....	16
2.2	运行项目.....	17
2.2.1	使用 Android 手机进行调测.....	18
2.2.2	使用 iPhone 手机或模拟器进行调测.....	21

2.2.3	修改 JSX 代码	22
2.2.4	ES 6 语法与 ES 5 语法	24
2.2.5	启动调试工具	25
2.3	构建注册页面	28
2.4	React Native 代码执行逻辑	32
2.5	UI 框架工作基本机制	33
2.5.1	状态机思维	33
2.5.2	“冒充常量”的状态机变量	35
2.5.3	“无处安放”的状态机变量	36
2.5.4	“努力瘦身”的状态机变量	36
2.6	React Native 组件间通信	37
2.7	深入理解 UI 重新渲染的过程	37
2.7.1	合并状态机变量	37
2.7.2	判断是否渲染	40
2.7.3	替换状态机变量	40
2.7.4	强制启动渲染	41
2.7.5	渲染过程	41
2.7.6	合并状态机变量的最简语法	41
2.8	React Native 组件的成员变量	42
2.9	React Native 组件的静态变量、静态函数	43
第 3 章	页面导航、弹出框及深入理解属性	44
3.1	分离注册组件、组件平台自适应	44
3.1.1	分离注册组件	44
3.1.2	组件平台自适应	44
3.1.3	平台检测	45
3.2	导航组件、挂接注册组件	45
3.3	挂接注册等待组件	47
3.4	Navigator 组件工作机制	49
3.4.1	push 与 pop	50
3.4.2	replace 函数	50
3.5	自定义组件	51
3.5.1	“弹出一切框”的实现	51
3.5.2	React Native 中颜色类型的值	53
3.5.3	挂接自定义组件	54
3.6	BackAndroid API 的 bug 与解决办法	56
3.7	属性确认	58
3.8	指定属性默认值	60

- 3.9 Alert 应用程序编程接口 60
 - 3.9.1 弹出确认框 60
 - 3.9.2 弹出选择框 61
- 3.10 带导航栏的页面导航 62
- 第 4 章 混合开发基础篇 63
 - 4.1 iOS 平台混合开发 63
 - 4.1.1 与 iOS 侧原生代码消息互通 64
 - 4.1.2 React Native 代码到 iOS 原生代码的消息 65
 - 4.1.3 iOS 原生代码到 React Native 代码的消息 68
 - 4.1.4 与 iOS OC 原生代码界面的切换 69
 - 4.1.5 应用初始界面设定 69
 - 4.1.6 iOS 混合开发中传递的参数类型 70
 - 4.1.7 混合开发中的多线程使用 70
 - 4.1.8 原生代码实现 Promise 机制 71
 - 4.1.9 跨语言常量 72
 - 4.2 Android 平台混合开发 73
 - 4.2.1 与 Android 原生代码消息互通 74
 - 4.2.2 React Native 代码到 Android 原生代码的消息 75
 - 4.2.3 与 Android 原生代码界面的切换 78
 - 4.2.4 Android 原生代码到 React Native 代码的消息 82
 - 4.2.5 应用初始界面设定 86
 - 4.2.6 传递的参数类型 86
 - 4.2.7 回调函数与 Promise 机制 86
 - 4.2.8 监听 ActivityResult 与 Android 生命周期事件 88
 - 4.2.9 混合开发中的多线程机制 89
 - 4.2.10 跨语言常量 89
- 第 5 章 flexbox 布局、View、Image 与可触摸组件 90
 - 5.1 flexbox 布局 90
 - 5.1.1 位置及宽、高相关样式键 91
 - 5.1.2 决定子组件排列规则的键 92
 - 5.1.3 决定组件显示规则的键 94
 - 5.1.4 边框、空隙与填充 95
 - 5.1.5 组件多样式声明与动态样式声明 96
 - 5.2 View 组件 97
 - 5.2.1 View 组件的颜色与边框 97
 - 5.2.2 View 组件的阴影与其他视觉效果 99

5.2.3	View 组件的变形	101
5.2.4	View 组件的回调函数	104
5.2.5	View 组件的其他属性	106
5.2.6	设备放置状态、根 View 与 onLayout 回调函数	106
5.2.7	pointerEvents 属性	109
5.3	Image 组件	111
5.3.1	加载网络图片	111
5.3.2	加载静态图片资源	112
5.3.3	加载资源文件中的图片	112
5.3.4	动态加载手机中的图片资源	112
5.3.5	Image 组件的样式	113
5.3.6	Image 组件显示特性	114
5.3.7	Image 组件的其他属性	117
5.4	可触摸组件	117
5.4.1	可触摸组件类型	118
5.4.2	TouchableOpacity 组件	118
5.4.3	TouchableHighlight 组件	118
5.4.4	其他属性	120
5.5	加深理解三大组件	120
5.5.1	使用导航栏的导航框架	121
5.5.2	等比放大无丢失显示图片	125
5.5.3	宽、高动态变化的组件呈现	128
第 6 章	Text、TextInput 等相关知识	129
6.1	Text 组件	129
6.1.1	样式键设置	129
6.1.2	其他属性	131
6.1.3	Text 组件的嵌套	131
6.1.4	文本显示的阴影效果	132
6.1.5	Text 居中显示	133
6.1.6	在字符串中插入图像	135
6.2	Text 组件在两个平台上的不同表现	136
6.2.1	只指定 fontSize, 不指定 height	137
6.2.2	只指定 height, 不指定 fontSize	137
6.2.3	fontSize 等于 height	137
6.2.4	height 大于 fontSize	138
6.2.5	边框在两个平台上的不同表现	138
6.3	TextInput 组件	140

6.3.1	TextInput 组件样式键	140
6.3.2	TextInput 组件的属性	140
6.3.3	TextInput 组件 iOS 平台专有属性	141
6.3.4	TextInput 组件 Android 平台专有属性	142
6.3.5	TextInput 组件的成员函数	142
6.4	TextInput 组件在两个平台上的不同表现	143
6.4.1	Android 平台的输入下画线	143
6.4.2	父组件的 alignItems 键失效	144
6.4.3	只指定 fontSize、不指定 height	145
6.4.4	height 等于 fontSize	145
6.4.5	height 大于 fontSize	146
6.4.6	边框在两个平台上的不同表现	146
6.5	TextInput 组件的生命周期	147
6.5.1	获得焦点	147
6.5.2	用户输入	147
6.5.3	用户按下提交键	147
6.5.4	失去焦点	148
6.6	软键盘与键盘事件	148
6.7	组件的引用	151
6.7.1	定义组件引用	151
6.7.2	得到系统定义的组件引用	151
6.7.3	调用组件的公开成员函数	152
6.7.4	重新设定组件的属性	152
6.7.5	获得组件的位置	154
6.8	跨平台状态栏组件	155
6.8.1	StatusBar 组件属性	155
6.8.2	StatusBar 组件使用示例	156
6.8.3	手机状态栏在开发中的处理	157
6.8.4	StatusBarIOS API	158
6.9	高度自增长的扩展 TextInput 组件	159
6.10	访问操作系统剪贴板	160
第 7 章	组件生命周期、数据存储及 React Native 应用实现步骤	163
7.1	组件生命周期	163
7.1.1	getInitialState	163
7.1.2	getDefaultProps	163
7.1.3	componentWillMount	164
7.1.4	componentDidMount	164

7.1.5	componentWillReceiveProps	164
7.1.6	shouldComponentUpdate	165
7.1.7	componentWillUpdate.....	165
7.1.8	componentDidUpdate.....	165
7.1.9	componentWillUnmount	166
7.2	读取 JSON 文件	166
7.3	数据持久化操作.....	167
7.3.1	flow 语法检查器.....	167
7.3.2	写入数据、错误捕捉	168
7.3.3	读取数据	170
7.3.4	AsyncStorage API 存储数据的无序性	173
7.3.5	删除数据	173
7.3.6	修改数据	175
7.3.7	JSON 对象存储	175
7.3.8	读取 JSON 对象	176
7.4	数据表操作.....	176
7.5	React Native 应用实现步骤、日记例程（上）	176
7.5.1	应用原型	177
7.5.2	基础组件结构设计	178
7.5.3	使用 React Native 组件搭建静态界面	180
7.5.4	React Native 组件分层	188
7.5.5	实现各组件业务逻辑	189
7.5.6	日记例程（上）总结	190
第 8 章	ScrollView 和 ListView	200
8.1	ScrollView 组件	200
8.1.1	ScrollView 组件属性.....	200
8.1.2	ScrollView 组件 iOS 平台专有属性.....	201
8.1.3	ScrollView 组件 Android 平台专有属性.....	202
8.1.4	ScrollView 组件的公开成员函数.....	203
8.1.5	RefreshControl 组件	203
8.1.6	ScrollView 组件基本用法	204
8.2	ListView 组件.....	206
8.2.1	ListView 组件的回调函数	207
8.2.2	ListView 组件的其他属性	208
8.2.3	ListView 组件的成员函数	208
8.3	简单的列表.....	208
8.3.1	准备列表的数据源	209

8.3.2	声明状态机变量	209
8.3.3	将数据源中的数据拷贝到 DataSource 中	210
8.3.4	定义如何渲染列表中的每一行	210
8.3.5	实现简单的列表	211
8.3.6	列表栏的高级处理	217
8.4	带分段标志的列表	217
8.4.1	准备数据源	218
8.4.2	声明状态机变量	218
8.4.3	将数据源中的数据拷贝到 DataSource 中	219
8.4.4	定义如何渲染每个分栏	219
8.4.5	定义如何渲染首、尾栏	220
8.4.6	列表间隔渲染	220
8.4.7	实现带分段标志的列表	220
8.4.8	总结	221
8.5	日记例程（下）总结	221
第 9 章	等待提示条、进度条和 Switch	222
9.1	ProgressBarAndroid 组件	222
9.1.1	ProgressBarAndroid 组件样式设置	222
9.1.2	ProgressBarAndroid 其他属性	222
9.1.3	Android 平台等待提示条	222
9.1.4	React Native 框架中定时器的使用	224
9.1.5	Android 平台进度条	225
9.2	iOS 进度条组件	226
9.2.1	ProgressViewIOS 组件样式设置	226
9.2.2	ProgressViewIOS 其他属性	226
9.2.3	iOS 平台进度条	226
9.3	iOS 平台等待提示条	227
9.3.1	ActivityIndicatorIOS 组件样式设置	227
9.3.2	ActivityIndicatorIOS 其他属性	227
9.3.3	iOS 平台等待提示条例程	227
9.4	Switch 组件	229
9.4.1	Switch 组件样式设置	229
9.4.2	Switch 其他属性	229
9.4.3	Switch 组件的使用	229
第 10 章	导航组件	231
10.1	导航组件的属性	231

10.1.1	回调函数	231
10.1.2	其他属性	232
10.2	导航器	232
10.3	NavigationBar	233
第 11 章	手势识别	240
11.1	PanResponder API	240
11.2	监视器	240
11.2.1	指定监视区域	241
11.2.2	定义监视器相关变量	241
11.2.3	准备监视器的事件处理函数	241
11.2.4	建立监视器	242
11.2.5	将监视器与监视区域挂接	242
11.3	监视事件的生命周期	242
11.3.1	单次点击事件的生命周期	243
11.3.2	单次点击事件处理	245
11.3.3	移动手势事件的生命周期	245
11.3.4	监视器异常事件	247
11.4	手势识别处理例程	247
11.4.1	单点手势——点击、拖动选择百分比参数	247
11.4.2	单点手势——带导槽的滑动来电接听或拒接界面	249
11.4.3	单点手势——滑动解锁屏幕界面	252
11.4.4	单点手势——单点任意方向拉动选择界面	254
11.4.5	两点手势	257
第 12 章	网络	258
12.1	获取网络状态	258
12.1.1	得到当前网络状态	258
12.1.2	监听网络状态改变事件	259
12.1.3	简单判断是否有网络连接	260
12.1.4	判断当前连接是否收费	260
12.2	通过 HTTP、HTTPS 与网络侧交换数据	260
12.2.1	发送请求	260
12.2.2	接收响应	263
12.3	在 React Native 开发中使用 AJAX 技术	264
第 13 章	网页浏览器、音视频媒体播放	266
13.1	WebView 组件样式设置	266

13.2	WebView 组件其他属性.....	266
13.2.1	非回调函数属性.....	266
13.2.2	回调函数属性.....	267
13.2.3	平台独有属性.....	267
13.2.4	WebView 组件成员函数.....	268
13.3	网页浏览器使用例程.....	268
13.3.1	浏览网页例程.....	268
13.3.2	加载本地网页例程.....	271
13.4	音视频媒体播放.....	273
第 14 章	图片的遍历、存取与显示	274
14.1	React Native 开发中 iOS 平台链接库的使用	274
14.2	获取手机中所有的图片信息.....	276
14.3	图片信息详解.....	278
14.3.1	Android 平台图片信息.....	278
14.3.2	iOS 平台图片信息.....	278
14.4	显示从 CameraRoll API 得到的图片	279
14.5	为用户提供图片选择界面.....	280
14.6	图片的保存与读取显示.....	282
14.6.1	保存图片数据.....	282
14.6.2	读取并显示图片	283
第 15 章	选择器、位置相关和应用状态	284
15.1	日期、时间选择器.....	284
15.1.1	DatePickerAndroid API	284
15.1.2	TimePickerAndroid API.....	286
15.1.3	DatePickerIOS 组件.....	287
15.2	Picker 组件	289
15.2.1	Picker 组件的样式设置.....	289
15.2.2	Picker 组件的属性.....	289
15.2.3	Picker.Item 组件属性	290
15.2.4	Picker 组件例程.....	290
15.3	PickerIOS.....	294
15.4	MapView 组件.....	295
15.4.1	MapView 组件样式设置	296
15.4.2	MapView 组件特有的跨平台属性	296
15.4.3	MapView 组件例程	297
15.5	AppState API	299