Ionic3 与 CodePush 初探

——支持跨平台与热更新的 App 开发技术

陈杰浩　张 成　吴 曦　史继筠 ◎ 编著

IONIC3 & CODEPUSH

—— APP DEVELOPMENT TECHNOLOGY SUPPORTING

CROSS PLATFORM & HOT UPDATE

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

需要全本请访问 www.ertongbook.com

Ionic3 与 CodePush 初探

——支持跨平台与热更新的 App 开发技术

陈杰浩　张 成　吴 曦　史继筠 ◎ 编著

IONIC3 & CODEPUSH

—— APP DEVELOPMENT TECHNOLOGY SUPPORTING

CROSS PLATFORM & HOT UPDATE

学院图书馆 藏书专用章

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

版权专有　侵权必究

图书在版编目（CIP）数据

Ionic3 与 CodePush 初探：支持跨平台与热更新的 App 开发技术 / 陈杰浩等编著 . —北京：北京理工大学出版社，2018. 4

ISBN 978 - 7 - 5682 - 5542 - 4

Ⅰ. ①I…　Ⅱ. ①陈…　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（2018）第 069254 号

出版发行 /北京理工大学出版社有限责任公司
社　　址 /北京市海淀区中关村南大街 5 号
邮　　编 /100081
电　　话 /（010）68914775（总编室）
　　（010）82562903（教材售后服务热线）
　　（010）68948351（其他图书服务热线）
网　　址 /http：//www. bitpress. com. cn
经　　销 /全国各地新华书店
印　　刷 /保定市中画美凯印刷有限公司
开　　本 /710 毫米 × 1000 毫米　1/16
印　　张 /20. 5
字　　数 /332 千字
版　　次 /2018 年 4 月第 1 版　2018 年 4 月第 1 次印刷
定　　价 /68. 00 元

责任编辑 /钟　博
文案编辑 /郭贵娟
责任校对 /周瑞红
责任印制 /王美丽

图书出现印装质量问题，请拨打售后服务热线，本社负责调换

前　言

App 的发展趋势

随着信息技术的快速发展，我们迎来了互联网时代，Web 应用逐渐取代了传统桌面应用的地位，构筑起整个互联网的主要骨架和内容，涉及国家、政府、企业以及社会的方方面面，可以说，我们的生活已经离不开 Web 应用。

移动设备的快速普及开启了全新的移动互联网时代，App 开发技术方兴未艾，Android8.0 与 iOS11 刚刚推送不久，全新的 Kotlin 与 Swift4 正式成为 Android 与 iOS 官方支持的开发语言，新一波技术浪潮滚滚而来。

技术的快速发展促成了 App 的诞生，然而编者认为，这只是前端表现形式上的变化，其未来的发展方向还是会回归到 Web 技术。如今市面上的很多原生 App 也在部分页面使用了 Web 技术，例如淘宝 App 中就引入了 Weex，以便更好地应对复杂多变的页面；也有很多 App 本身完全通过 Web 技术实现，例如微信小程序就是一种大胆的尝试，其屏蔽了 Android 与 iOS 底层的技术细节。

由此不难看出，App 开发依然可以通过 Web 技术实现，通过 Web 技术实现的 App 有时甚至可以具备原生 App 所不具备的优势，编者认为这也是

未来 App 的发展趋势。为了帮助更多有志于 Web 开发的人，编者决定编写此书，力图让读者利用已掌握的 Web 技术完成 App 的开发工作，为信息化事业的发展贡献力量，这也是编者及其所在团队的初心。

机遇与挑战并存

不论是全新的 Kotlin 与 Swift4，还是传统的 Java 与 Object - C，其开发出的 App 都属于原生 App，这类 App 一直面临着两大难题：一是无法做到跨平台；二是难以做到热更新。

原生 App 无法做到跨平台。在应用的开发阶段，开发者需要为 Android、iOS、Windows 等平台进行多次编码，且不同平台要求开发者具备不同的开发技术，这使开发成本居高不下。在应用的维护阶段，修改任何界面或功能，都需要同时修改不同平台的不同代码，这也使维护成本很高。

原生 App 难以做到热更新。每次更新 App 时，都需要编译生成新的二进制安装文件，并且需要提交到应用商店等待审核。这显然拖慢了原生 App 的更新速度，更使对严重缺陷的紧急修复变得十分被动。

针对以上两个问题，Ionic3 可以解决跨平台问题，CodePush 可以解决热更新问题。Ionic3 是一个基于 HTML5 与 Cordova 的开发框架，并且具备仿原生的精美控件，CodePush 是一项代码推送技术，可以将代码以热更新包的形式直接推送到用户终端。Ionic3 于 2017 年 4 月 5 日正式发布，CodePush 至今依然处于发布预览状态，因此这两项技术都是比较新的技术，为 App 开发带来了新的机遇与挑战。

说其是机遇，是因为 Ionic3 与 CodePush 可以显著降低 App 的开发成本，甚至可以取代中小型的原生 App，并且背后都有大公司提供技术支持，有比较乐观的发展前景。说其是挑战，是因为新的技术势必会带来一定的学习成本，尤其是 Ionic3 的学习曲线略显陡峭，甚至会颠覆一些传统的开发思维。

iOS 热更新的可行性

苹果公司近期宣布封杀热更新技术，并且下架了一批不符合要求的 App，这使业界普遍认为热更新技术已经走到了尽头，但是这种想法是十分错误的，原因如下：

（1）苹果公司确实限制了一部分热更新技术，主要涉及一些可以影响原生代码执行的技术，本书所使用的 CodePush 技术完全基于 HTML5，因此不可混为一谈。

需要全本请访问 www.ertongbook.com

（2）CodePush 官网上很早以前就对这个问题进行过说明，详见 http://microsoft.github.io/code-push/faq/index.html。

（3）苹果开发者计划许可协议（Apple Developer Program License Agreement）3.3.2 中明确说明了允许基于 HTML5 的热更新技术，即便是原生 App，也可以通过 WebView 控件（App 中调用浏览器内核渲染 Web 内容的控件）实现，这是再正常不过的现象。

（4）编者在 App Store 上架的 App 并没有收到过任何警告，也没有被苹果公司下架，并且在此次风波中还多次提交过更新，均顺利通过审核。

本书的适用范围

本书需要读者有一定的编程基础和开发经验。由于 Ionic3 是一个 HTML5 开发框架，因此需要读者具备基本的 Web 开发知识，包括 HTML、CSS、JavaScript 的相关知识。Ionic3 使用到了 TypeScript 这门崭新的语言，这是一门具备面向对象特性的语言，因此还需要读者具备基本的面向对象的开发思想。

本书参考了各类技术的官方文档，筛选出了核心知识点，在保留官方经典示例的基础上，用更加通俗易懂的语言重新讲解了一遍。Ionic3 与 CodePush 的官方文档均为英文，本书并没有直接翻译相关文档，而是在确保知识点正确的前提下，用中文重新梳理了一遍。官方文档的特点是全面而繁杂，本书的特点是浅显而精简，因此本书更适合渴望快速入门的初学者。

本书只针对 Ionic3 与 CodePush 的基础知识进行了讲解，省略了很多不常用或难度较大的进阶内容，如果读者想更加深入地学习相关知识，书中也提供了进一步学习的相关建议。

本书的章节结构

本书共有 14 章，可以分为三大部分。

第一部分包含第 1 章～第 3 章的内容，主要介绍了相关技术以及环境的搭建，让读者在还没有掌握充足的知识前也能运行一个示例项目，力求给读者一个感性的认识。

第二部分包含第 4 章～第 12 章的内容，按顺序详细讲解了各项技术，借鉴了官方文档的内容并且作了大幅精简，力求让读者在最短的时间内掌握核心实用的内容。

第三部分包含第 13 章～第 14 章的内容，综合运用前面章节的知识设计并实现了一款示例 App，并且包含上架应用商店的流程，力求让读者有

一个可供模仿借鉴的例子。

本书包含 Ionic3 与 CodePush 两方面的内容，虽然从逻辑上来说，跨平台与热更新是两个对等的概念，但从技术角度来看，Ionic3 的相关内容至少占据 80% 的篇幅，CodePush 的相关内容最多只能占据 20% 的篇幅。

读者有必要知悉，Ionic3 可以理解为一种开发框架，读者在掌握基础知识后依然有十分巨大的发挥空间，需要依靠长期的项目经验才能摸索出最佳的实践方法。CodePush 的本质是一个第三方库，读者在熟悉了基本概念和常用 API 之后，就可以在任何一款 App 中反复使用了。

致谢

首先感谢许昌达，是他探索并发现了 Ionic 跨平台开发框架，并且在早期提供了很多必要的技术支持，也最终促成了本书的诞生。

其次感谢参与本书校稿工作的人员：郑泉斌、张秋鸿、过其靖、王晓华、李弋豪、赵子芊，正是他们的辛勤付出，使本书能以更高的质量与读者见面。

其中还要特别感谢郑泉斌，作为一个零基础的人，他完整阅读了本书的草稿版本，以初学者的视角对本书提出了很多宝贵的修改建议。

编者寄语

到目前为止，国内有关 Ionic3 与 CodePush 的资料依然非常稀少，官方英文文档的门槛导致这两项技术在国内依然属于小众技术。市面上已经出版发行的相关书籍屈指可数，有些仍在介绍第一代 Ionic 的内容，这与当今最新的 Ionic3 相比已经明显过时。

编者拥有将 Ionic3 与 CodePush 用于实际项目的经验，并且获得了比较理想的效果，因此希望将自己的探索经验分享给大家，以起到抛砖引玉的作用，为 App 的开发和维护提供一种新的思路和方向。

由于编者水平有限，错漏之处在所难免，敬请广大读者批评指正。Ionic 与 CodePush 发展较快，因此实际情况可能会与书中的内容有所出入，如果读者在阅读本书的过程中遇到任何问题，欢迎与我们取得联系（908544864@qq.com），我们保证知无不言，言无不尽，希望大家一起探讨，共同进步！

编　者

于北京理工大学

目　录

第 1 章　相关技术介绍 …… 1

1.1　探究开发模式 …… 1

1.1.1　Native App …… 1

1.1.2　Web App …… 2

1.1.3　Hybrid App …… 2

1.1.4　React Native …… 3

1.1.5　微信小程序 …… 4

1.2　梳理学习路线 …… 4

1.2.1　依赖关系 …… 4

1.2.2　TypeScript …… 5

1.2.3　Angular4 …… 6

1.2.4　Ionic3 …… 7

1.2.5　Cordova …… 8

1.2.6　Ionic Native …… 10

1.2.7　CodePush …… 10

第 2 章 配置开发环境 …… 12
2.1 安装 Node. js …… 12
2.2 使用淘宝 NPM 镜像 …… 14
2.3 安装 Ionic CLI …… 14
2.4 安装 CodePush CLI …… 15
2.5 安装 Visual Studio Code …… 16
2.6 配置 Android 环境 …… 19
2.6.1 Java 环境变量 …… 19
2.6.2 Android Studio …… 19
2.6.3 Android 模拟器 …… 20
2.7 配置 iOS 环境 …… 23
2.7.1 前期准备 …… 23
2.7.2 Xcode …… 23
2.7.3 iOS 模拟器 …… 24
第 3 章 Ionic3 初体验 …… 26
3.1 新建工程项目 …… 26
3.2 浏览器的调试 …… 27
3.3 模拟器的调试 …… 28
3.3.1 Android 模拟器的调试 …… 28
3.3.2 iOS 模拟器的调试 …… 30
3.4 梳理目录结构 …… 32
3.5 查看官方 Demo …… 34
第 4 章 TypeScript 基础语法入门 …… 36
4.1 声明与建议 …… 36
4.2 配置独立环境 …… 36
4.3 变量类型 …… 37
4.3.1 类型定义 …… 37
4.3.2 类型推断 …… 39
4.3.3 类型断言 …… 39
4.4 函数 …… 39
4.4.1 类型限定 …… 39
4.4.2 可选参数 …… 40
4.4.3 默认参数 …… 41
4.4.4 剩余参数 …… 41

4.4.5 箭头函数 …… 42
4.5 类 …… 43
4.5.1 面向对象 …… 43
4.5.2 继承 …… 43
4.5.3 存取器 …… 44
4.5.4 其他特性 …… 45
4.6 接口 …… 45
4.6.1 类型限定 …… 45
4.6.2 实现 …… 46
4.6.3 多继承 …… 47
4.7 模块 …… 47
4.7.1 基本概念 …… 47
4.7.2 导出 …… 48
4.7.3 导入 …… 49
4.8 其他 …… 50
4.8.1 模板字符串 …… 50
4.8.2 迭代器 …… 51
4.8.3 装饰器 …… 51
第5章 Angular4 核心思想简介 …… 53
5.1 声明与建议 …… 53
5.2 配置独立环境 …… 54
5.3 整体架构 …… 55
5.3.1 全新理念 …… 55
5.3.2 “八大金刚” …… 55
5.4 模块（Module） …… 57
5.4.1 根模块 …… 57
5.4.2 对比 TypeScript 模块 …… 58
5.4.3 常用模块 …… 58
5.5 组件（Component） …… 59
5.5.1 封装复用 …… 59
5.5.2 局部样式 …… 61
5.5.3 输入/输出 …… 62
5.5.4 函数调用 …… 63
5.5.5 层次结构 …… 64

5.6 模板（Template） …… 65
5.6.1 模板语法 …… 65
5.6.2 插值表达式 …… 66
5.6.3 模板引用变量 …… 66
5.6.4 管道 …… 67
5.6.5 安全导航操作符 …… 68
5.7 数据绑定（Data Binding） …… 68
5.7.1 基本概念 …… 68
5.7.2 属性绑定 …… 69
5.7.3 CSS 类绑定 …… 69
5.7.4 CSS 样式绑定 …… 70
5.7.5 事件绑定 …… 70
5.7.6 双向数据绑定 …… 71
5.8 指令（Directive） …… 72
5.8.1 指令分类 …… 72
5.8.2 ngClass …… 72
5.8.3 ngStyle …… 73
5.8.4 ngIf …… 73
5.8.5 ngFor …… 74
5.8.6 ngSwitch …… 75
5.9 服务（Service） …… 76
5.9.1 职责分离 …… 76
5.9.2 依赖注入 …… 76
5.9.3 Promise …… 78
5.9.4 网络服务 …… 80
第 6 章 Ionic3 页面布局控件 …… 83
6.1 声明与建议 …… 83
6.2 顶栏与底栏（Header & Footer） …… 84
6.3 导航栏（Navbar & Toolbar） …… 84
6.4 内容（Content） …… 86
6.5 滚动（Scroll） …… 89
6.6 滑动（Slide） …… 89
6.7 菜单（Menu） …… 93
6.7.1 菜单组件 …… 93

需要全本请访问 www.ertongbook.com

6.7.2 菜单服务 …… 95
6.8 网格（Grid & Row & Col） …… 98
6.8.1 十二列布局 …… 98
6.8.2 屏幕自适应 …… 100
6.9 标签（Tab） …… 102
6.9.1 文字标签 …… 102
6.9.2 图标标签 …… 103
6.9.3 文字图标标签 …… 104
6.9.4 徽章标签 …… 104
6.9.5 标签组 API …… 105
6.9.6 标签 API …… 107
6.10 段（Segment） …… 107
6.11 分裂板（SplitPane） …… 110
第7章 Ionic3 列表相关控件 …… 112
7.1 列表（List） …… 112
7.1.1 普通列表 …… 112
7.1.2 无线列表 …… 113
7.1.3 内联列表 …… 114
7.1.4 分组列表 …… 115
7.1.5 列表头 …… 116
7.2 条目（Item） …… 117
7.2.1 布局位置 …… 117
7.2.2 滑动条目 …… 119
7.2.3 重排序 …… 120
7.2.4 其他 …… 122
7.3 注释（Note） …… 123
7.4 下拉刷新（Refresher） …… 124
7.5 上拉加载（InfiniteScroll） …… 127
7.6 虚拟滚动（VirtualScroll） …… 129
第8章 Ionic3 弹出层控件 …… 130
8.1 动作菜单（Action Sheet） …… 130
8.2 对话框（Alert） …… 133
8.2.1 基本对话框 …… 133
8.2.2 输入对话框 …… 135

8.2.3 确认对话框 …… 137
8.2.4 单选对话框 …… 139
8.2.5 复选对话框 …… 140
8.3 加载框（Loading） …… 142
8.4 模态框（Modal） …… 144
8.5 浮动框（Popover） …… 146
8.6 弹出框（Toast） …… 148
第 9 章 Ionic3 页面元素控件 …… 151
9.1 头像（Avatar） …… 151
9.2 徽章（Badge） …… 152
9.3 按钮（Button） …… 153
9.3.1 默认样式 …… 153
9.3.2 轮廓样式 …… 154
9.3.3 简易样式 …… 155
9.3.4 圆形按钮 …… 156
9.3.5 块状按钮 …… 157
9.3.6 全宽按钮 …… 157
9.3.7 按钮尺寸 …… 158
9.3.8 图标按钮 …… 159
9.4 卡片（Card） …… 160
9.4.1 简易卡片 …… 160
9.4.2 复合卡片 …… 161
9.5 复选按钮（Checkbox） …… 163
9.6 碎片（Chip） …… 165
9.7 日期时间（DateTime） …… 167
9.8 浮动按钮（FAB） …… 171
9.9 输入（Input） …… 173
9.9.1 普通输入 …… 174
9.9.2 固定输入 …… 175
9.9.3 栈式输入 …… 176
9.9.4 浮动输入 …… 177
9.9.5 相关属性 …… 178
9.10 单选按钮（Radio） …… 178
9.11 范围（Range） …… 180

9.12 搜索栏（Searchbar） …… 182
9.13 选择（Select & Option） …… 184
9.14 开关按钮（Toggle） …… 186
9.15 缩略图（Thumbnail） …… 187
9.16 文字样式（Typography） …… 188
第 10 章 Ionic3 其他知识点 …… 189
10.1 配置文件 …… 189
10.1.1 根模块 …… 189
10.1.2 config. xml …… 191
10.1.3 index. html …… 193
10.2 页面生命周期 …… 196
10.2.1 基本概念 …… 196
10.2.2 流程详解 …… 196
10.2.3 根组件 …… 199
10.3 页面导航 …… 200
10.3.1 基本概念 …… 200
10.3.2 根组件导航 …… 201
10.3.3 页面组件导航 …… 202
10.3.4 标签（Tab）导航 …… 204
10.3.5 NavController API …… 206
10.4 全局事件 …… 207
10.5 全局配置 …… 209
10.6 数据持久化 …… 212
10.7 主题样式 …… 214
10.7.1 分层控制 …… 214
10.7.2 SCSS 简介 …… 215
10.7.3 主题颜色 …… 218
10.7.4 样式覆盖 …… 219
10.8 字体图标 …… 220
第 11 章 Cordova 插件与 Ionic Native …… 222
11.1 完善跨平台体验 …… 222
11.2 Cordova 插件库 …… 223
11.3 安装 Cordova 插件 …… 225
11.3.1 下载安装 …… 225

11.3.2 使用说明 …… 226
11.3.3 模拟测试 …… 228
11.4 卸载 Cordova 插件 …… 229
11.5 更新 Cordova 插件 …… 229
11.5.1 更新检测工具 …… 229
11.5.2 更新操作步骤 …… 230
11.6 不使用 Ionic Native …… 231
11.6.1 特殊调用方式 …… 231
11.6.2 改造回调函数 …… 232
第 12 章 CodePush 集成与使用 …… 234
12.1 声明与建议 …… 234
12.2 引入热更新机制 …… 234
12.3 开发端 CLI …… 235
12.3.1 基本概念 …… 235
12.3.2 注册账号 …… 236
12.3.3 App 管理 …… 237
12.3.4 分支管理 …… 239
12.3.5 推送更新 …… 240
12.3.6 版本控制 …… 241
12.3.7 提升操作 …… 242
12.3.8 回滚操作 …… 243
12.3.9 历史记录 …… 244
12.3.10 统计数据 …… 244
12.4 客户端 SDK …… 245
12.4.1 基本概念 …… 245
12.4.2 下载安装 …… 246
12.4.3 自动同步 …… 247
12.4.4 其他 API …… 250
12.5 Mobile Center …… 251
第 13 章 示例 App 设计与实现 …… 253
13.1 示例说明 …… 253
13.2 前期准备 …… 254
13.3 Ionic3 页面封装 …… 257
13.3.1 设计思想 …… 257

13.3.2 列表页面 …… 258
13.4 Ionic3 组件封装 …… 266
13.4.1 设计思想 …… 266
13.4.2 条形菜单组件 …… 266
13.4.3 图片选择组件 …… 268
13.5 Ionic3 服务封装 …… 273
13.5.1 设计思想 …… 273
13.5.2 对话框服务 …… 273
13.5.3 网络交互服务 …… 275
13.6 集成 CodePush …… 284
13.6.1 服务封装 …… 284
13.6.2 自动更新 …… 290
13.6.3 分支切换 …… 291
第 14 章 发布到应用商店 …… 293
14.1 性能优化策略 …… 293
14.1.1 性能瓶颈 …… 293
14.1.2 Crosswalk …… 294
14.1.3 WKWebView …… 295
14.1.4 预编译与摇树优化 …… 296
14.2 安全加固策略 …… 297
14.2.1 安全风险 …… 297
14.2.2 代码压缩与代码混淆 …… 297
14.2.3 签名机制 …… 298
14.3 打包并提交 …… 300
14.3.1 Android 生成 apk …… 300
14.3.2 iOS 生成 ipa …… 301
14.4 更新与维护 …… 302
14.4.1 选择更新方式 …… 302
14.4.2 更新 Ionic …… 303
结语 …… 306
参考文献 …… 309

第1章 相关技术介绍

1.1 探究开发模式

1.1.1 Native App

Native 即原生，是传统的开发模式，针对各个移动平台需要采用特定的编程语言，比如 Android 平台需要使用 Java，iOS 平台需要使用 Swift，Windows 平台需要使用 C#。Native App 可以调用相应平台的全部原生 API，原生 API 是各移动平台官方提供的 API，比如调用摄像头拍照、调用麦克风录音、调用振动器振动、调用陀螺仪获取 3D 动作等。因此，Native App 可以实现最丰富的功能，同时也具有最佳的性能。

但是，这种开发模式需要针对多个平台分别开发相应的 App，不具备跨平台能力，修改任何代码都需要重新编译生成新的二进制安装文件，也不具备热更新能力。

目前市面上的绝大多数 App 都属于 Native App，尤其是那些超大型的

需要全本请访问 www.ertongbook.com