

O'REILLY®



基于Ionic的 移动App开发

Mobile App Development with Ionic 2

Chris Griffith 著
杨宏焱 译

力出版社

基于Ionic的移动App开发

Chris Griffith 著

杨宏焱 译

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

Copyright © 2017 Christopher Griffith. All rights reserved.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2017.
Authorized translation of the English edition, 2017 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2017。

简体中文版由中国电力出版社出版 2017。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

图书在版编目 (CIP) 数据

基于Ionic的移动App开发 / (美) 克里斯·格里菲斯 (Chris Griffith) 著；杨宏焱译. — 北京：中国电力出版社，2017.12

书名原文：Mobile App Development with Ionic 2

ISBN 978-7-5198-1424-3

I. ①基… II. ①克… ②杨… 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291696号

北京市版权局著作权合同登记 图字：01-2017-5675号

出版发行：中国电力出版社

地 址：北京市东城区北京站西街19号（邮政编码100005）

网 址：<http://www.cepp.sgcc.com.cn>

责任编辑：刘 焱 (liuchi1030@163.com)

责任校对：朱丽芳

装帧设计：Karen Montgomery, 张 健

责任印制：藺义舟

印 刷：三河市百盛印装有限公司

版 次：2017年12月第一版

印 次：2017年12月北京第一次印刷

开 本：750毫米×980毫米 16开本

印 张：18.5

字 数：351千字

印 数：0001—3000册

定 价：68.00元

版 权 专 有 侵 权 必 究

本书如有印装质量问题，我社发行部负责退换

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

目录

序	1
前言	5
第 1 章 混合移动 App	11
Ionic 框架是什么?	12
Ionic 2 有什么新特性?	12
不同移动开发方式的比较	13
理解 Ionic 的技术栈	15
进行 Ionic 应用程序开发的必备条件	17
小结	18
第 2 章 配置开发环境	19
安装 Ionic 框架	19
新建 Ionic 项目	22
安装平台工具	25
配置模拟器	26
配置你的设备	28
添加移动平台	31
在模拟器上测试	31
在设备上测试	32
小结	33

第 3 章 理解 Ionic 命令行界面	34
指定编译平台	36
管理 Cordova 插件.....	37
Ionic 生成器	38
预览你的 App	38
指定 IP 地址	39
模拟运行 Ionic App	40
在设备上运行 Ionic App.....	42
输出日志	42
CLI 的信息	42
小结	43
第 4 章 Angular 和 TypeScript 基础	44
Angular 2 是什么?	44
理解 ES6 和 TypeScript	54
类型化函数.....	60
小结	60
第 5 章 Apache Cordova 基础.....	61
Cordova（即 PhoneGap）历史.....	63
Apache Cordova 与 Adobe PhoneGap	63
深入了解 Cordova	64
配置你的 Cordova App	65
设备的可访问性（即插件）	65
界面组件：缺失的拼图	66
为什么不使用 Cordova.....	66
理解 Web 标准.....	66
小结	68
第 6 章 理解 Ionic	69
HTML 的构成.....	69
Ionic 组件	71
理解 SCSS 文件.....	71

理解 TypeScript	73
小结	74
第 7 章 编写我们的 Ionic2Do App	75
添加平台	76
预览 Ionic2Do App	76
修改页面结构	87
添加全扫手势	98
简单主题	99
正确地声明类型	100
保存数据	101
创建 FireBase 账号	101
安装 Firebase 和 AngularFire 2	102
Ionic 编译系统	103
将 AngularFire 添加到 app.module.ts 文件	105
使用 Firebase 数据	105
使用 Ionic Native	108
小结	112
第 8 章 创建一个基于 Tab 的 App	113
引导我们的 App	118
通过 HTTP 服务加载数据	119
显示我们的数据	122
生成新页面	125
理解 Ionic 2 的导航模型	126
修改公园详情页	128
渲染 Google 地图	131
添加大头钉	133
添加查找功能	137
设置 App 的样式	140
虚拟滚动	142
定制表格 header	144
小结	146

第 9 章 构建一个天气应用	147
开始	147
探究侧滑菜单模板	149
侧滑菜单选项	152
显示菜单	152
转换模板	153
模拟天气提供者	156
显示天气数据	158
进度显示：loading 对话框和下拉刷新	161
添加 Geolocation	163
访问在线天气数据	165
将 Geolocation 和 Weather 提供者关联	166
获取其他地区的天气	169
下拉刷新：第二部分	171
编辑地址	172
使用 Geocoding 服务	177
动态刷新侧滑菜单	180
Ionic 事件	181
Observable	183
调整 App 的样式	187
添加天气图标	191
下一步	194
小结	195
 第 10 章 调试并测试你的 Ionic 应用	 196
解决 CORS 问题	201
用 iOS 或 Android 模拟器运行	202
在设备上进行调试	203
调试 Ionic 的初始化	204
其他工具	205
小结	206

第 11 章 部署你的应用程序	207
修改 config.xml 文件	207
App 图标和 splash 图片	208
编译你的 Android APK	208
编译你的 iOS App	211
小结	225
第 12 章 探索 Ionic Cloud	226
创建 Ionic Cloud 账号	226
生成你的 Ionic App ID	227
配置你的 App	227
Ionic 部署	228
安全文档	236
Ionic 打包	242
Ionic View	245
Ionic Creator	248
小结	249
第 13 章 渐进式 Web 应用	250
但是，什么是渐进式 Web App ?	251
manifest.json 文件	252
推送通知	256
小结	257
第 14 章 终章	258
下一步	265
Ionic 论坛	265
小结	266
附录 A 从 Ionic 1 升级到 Ionic 2	267
附录 B 理解 config.xml 文件	270
附录 C Ionic 组件库	280

在 2013 年，我们小组正忙于几个拖曳式工具的开发，用在两个最流行的移动和桌面 Web 框架上：jQuery Mobile 和 Bootstrap。我们发现在 Web 开发中的可重用组件和框架发生了快速增长，正在创建更好和更具包容性的工具，让它们的使用变得更加简单。

刚好在那个时候先后发布了 iPhone 5 和 iOS 7，Web 性能得到了明显提升，新的 Web API 也开放了，移动浏览器 App 的性能和特性达到了前所未有的程度。我们开始设想，能不能利用这些新特性搭建出一个框架，提供类似原生一样的 UI 工具包，让 Web 开发者能够用标准的浏览器技术创建出原生品质的 App？你可能会说，这是“移动版的 Bootstrap”吧？

恰逢此时，Angular 1 被广泛使用于各种 Web 开发环境，完美地解决了 JavaScript 和 HTML Web 组件的重用问题。我们决定搭建一个移动优先的 Web UI 框架，使用高速增长的 Angular 1 框架，来实现交互性和可分发。

Ionic 的第一版发布于 2013 年末，它吸引了 Web 开发者的兴趣，这个项目使其在 GitHub 上的 star 数目和用 npm 安装的数目快速增长。一年半后，创业公司、开发团队和企业用户用这个项目开发超过 100 万个 App。

2015 年，JavaScript 在一夜之间发生了翻天覆地的变化。ES 5，这个从 Web 2.0 时代就广为人知的 JavaScript 版本变成了过去式。取而代之的是 ES 6，新一代的 JavaScript 版本产生了，具有全新的面向对象开发特性，如新的 sharing 和

loading 模块，更简洁的语法等。整个 JavaScript 世界因为浏览器运行时沸腾不已，开发者争先恐后地用上了 ES 6。

转译器被编写出来了，用于将新的 JavaScript 语法转译成 ES 5 语法以便浏览器能够理解。开发者正在摸索将 JavaScript 库分发为可重用模块的最佳做法。新的用于编译和发布独立 JavaScript 模块的构建工具不断被发明、抛弃和重建。试图减少错误并使现代 JavaScript 语法更加标准化的新项目出现了，比如 TypeScript 和 Flow。随着 ES 7 以后的实验特性不断被添加到转译器中，在新特性进入生产代码库之前，那些被抛弃和从标准中删除的特性会让保守的 JavaScript 开发者更加懊悔。总之，这是一场混乱。

来自 ES 6 之前时代的框架作者，突然会面临着一个令人生怯的任务，将自定义的抽象换成标准的 ES 6 以后有效的抽象。这些框架，很少有动力开发出像 Angular 1 一样的抽象。对于 Angular 来说，问题十分简单：如何将所有的和框架相关的东西比如 scope、controller、directive 等转换成标准的 JavaScript 类、Web 组件一类的东西呢？

通过 JavaScript 的这次罕见的升级，Angular 团队从第一个主要的 JavaScript 框架的构建中取得了一些经验，并在未来的兼容 Web 和移动开发的框架中应用这些经验。这并不妨碍将 Angular 1 的主要概念映射成非常成熟的 ES 6 概念。事实上，从许多方面来说，它们在 ES 6 中变得更自然。

当得知 Angular 2 要开发出来的时候，我们立即意识到这是一个机会，将从 Ionic 1 的构建以及那 100 万 App 的制作中得到的经验用到新的框架中来。

Ionic 团队从 2015 年春开始编写 Ionic 2。经过一年半的开发，犯了一些错，找到了新的解决方案，进行了大胆的尝试，我们最终兴奋地推出了一个大版本，可用于生产的 Ionic 2。

从表面上看，Ionic 2 和 Ionic 1 一样。组件使用自定义 HTML 标签编写，这些标签被 Ionic 转换成强大的移动组件。Action 被绑定到类的回调上，这个类扮演着 App 中某个页面的控制器角色。项目的编译和测试使用的是相同的命令行工具。外观和主题是类似的，Ionic 2 的外观和 Ionic 1 别无二致。

不同的地方在于底层。Ionic 2 底层是用 TypeScript 和 Angular 2 重写的。所有的 Ionic 代码都是类型化的，在我们自己的代码中明显降低了 bug 和类型问题的出现几率。还有一些让人惊喜的新特性，比如，对于使用 Atom 和 Visual Studio Code 的开发者而言，可以使用行内文档并能轻松重构了。代码更加面向对象了，特别是对于一个 UI 框架来说，尤其有用。结构不再像 Angular 1 那么原始。

Angular 2 为实现在手机上平滑运行进行了重构，采用了一系列手段，比如降低性能消耗，使核心操作（比如变化感知）变得流畅。因此，Ionic 2 App 运行起来比 Ionic 1 App 更快，能够处理更复杂的任务。

Ionic 的目的一直是使用最轻松的方法构建优秀的移动应用。如果我们不能确信在提升功能的同时使 Ionic 变得更易用，我们是不会花费时间和成本重写框架的。我们相信 TypeScript 能够使 Ionic 代码书写和理解起来都更加轻松。我们相信 Angular 2 比 Angular 1 更加容易使用，它需要的域特定语言和知识要少得多。我们相信 Ionic 2 项目更简洁，组织性更好，组件的使用更加简单。

除了底层技术，Ionic 2 还有几个主要的新特性。现在，Ionic 2 通过对材料设计的支持和简化的主题样式，让 App 的外观和运行时的平台相适配。我们的导航系统能够创建出各种灵活的、与原生 App 类似的导航，而这是在浏览器中无法做到的。我们添加了太多的特性、组件和数不清的原生 API。

另外，移动领域在 2016 年还发生了一个剧烈变化。突然之间，移动 Web 又重新流行起来，同时渐进式 Web App 以一种强势的方式进入前台。Google 打通了一个全新的世界，App 运行在浏览器中，不需要安装，不管网络带宽和网络是否连接，都能够提供良好的用户体验，移动开发者对于将移动 Web 作为他们的移动解决方案深信不疑。

使用 Ionic 2 的开发者现在可以在不修改任何代码的情况下迁移到移动 Web。Ionic App 既是一个运行在 iOS 和 Android 设备上的原生 App，也是一个运行在移动 Web 上的渐进式 Web App。一次编写，随处运行。

我们将全部重心放在 Ionic 2，很高兴能够将 Ionic 2 推荐给可用于生产的移动 App。我们希望你能够和我们一样发现它是一个高性能的、灵活的框架，用它编写移动 App 和移动网站比你想象的还要容易。有接近三百万的 App 是用 Ionic

构建的。我们对如何搭建高质量的 App 框架略知一二，汲取了每一个经验教训，并将它们全部用到了 Ionic 2 中。

如果你喜欢 Ionic 2，希望你也了解一下我们编写的配套工具，它们能给 Ionic 开发者提供一些便利，包括快速测试工具 Ionic View、快速原型创建工具、快速 App 开发工具 Ionic Creator，以及在 Ionic Cloud 中的一整套轻量级后端集成服务。Ionic 正在变成一个和移动开发相关的“一站式”商店。

我代表全体 Ionic 团队成员，预祝你 Ionic 2 用得开心，让我们在论坛 (<https://forum.ionicframework.com>) 中再见！

——Max Lynch
创始人 / CEO, Ionic

前言

我第一次涉足移动开发领域的事情要从 2007 年说起。当时，我正在应聘 Qualcomm 用户体验团队中的 UX 工程师职位，恰逢史蒂夫·乔布斯发布他的第一台 iPhone。我的面试过程几乎变成了一场苹果发布会的讨论。那一天所发生的一切影响了我的整个职业生涯。接下来的十年间，我接触过各种移动开发解决方案。一直以来，我的最终目标是找到一种能够快速开发、让团队能够进行快速测试和快速实践新点子及想法的解决方案。

我曾经做过许多移动原型设计，它们的用户体验都是高度定制化的。需要模仿真实设备原生组件的需求几乎没有。偶尔需要用到原生组件的时候，我都是在方案中用到时重新设计。到最后，原型设计中定制化的成分越来越少，大部分都采用默认设计。我需要找出一种能够提供大量组件的解决方法，因为我不再想自己开发和维护某个定制组件了。

我开始尝试在一些项目中使用 Ratchet 和 TopCoat 这样的库。我在加利福尼亚大学的“圣地亚哥推广计划”中给学生们上一门课叫做“移动开发入门”，是基于 jQuery Mobile（和 PhoneGap Build）的。但是，这些解决方案都不能在创建原型时为我提供丰富的组件库。

我不记得什么时候知道了 Ionic 框架，但我知道它是基于 Apache Cordova 和 AngularJS 的。我曾经录制过两个关于 PhoneGap Build 和 Apache Cordova 的视频课程，但对 AngularJS 了解得非常少。由于原型开发的原因，我通常会对那些大框架望而生怯。不久以后，我看到别的组件库也使用了相同的技术栈。因此，

我决心开始学习 AngularJS 和 Ionic 框架。很快，我就被这两个框架所展示出来的威力所折服，开始在我的解决方案中使用它们。

我开始疯狂学习 Ionic，直到我发布了第一个用 Ionic 编写的商业移动应用 Hiking Guide:Sedona。随后，在 2015 年 10 月，Ionic 2 发布了。这次发布不仅仅是一次升级，同时也是一次重大突破。因此同样的过程开始了：学习新的 Angular 语法，使用 ES 6、TypeScript，甚至我将原来的编辑器升级到了微软的编辑器！在这个过程中 Ionic 2 也在不断成长和趋于成熟。

编写本书是一个漫长和有趣的过程。Ionic 的每个版本的发布都会迫使我认真阅读它的修改日志，评估它对已写和未完成章节的影响。这些都会让我对这个框架有更深入的理解。我希望本书成为一本学习使用 Ionic 编写混合移动应用的指南。

Chris Griffith, 圣地亚哥, 2017 年 1 月

本书面向的读者

本书适合准备学习 Ionic 框架的初学者。本书需要你熟悉 JavaScript、HTML 和 CSS。本书会涉及一些 TypeScript、ES 6、Angular 2 和 Apache Cordova 的主要概念，这些内容你可以参考更多相关资源。本书是以 step by step 方式编写的，请放心阅读并学习如何用 Ionic、Angular 和 Cordova 编写混合移动 App 吧。

本书结构

本书会带你依次了解 Ionic 框架的每一部分。各章内容简要介绍如下：

- 第 1 章，混合移动 App，介绍混合移动应用的概念。
- 第 2 章，配置我们的开发环境，介绍编写 Ionic 应用程序需要些什么。
- 第 3 章，理解 Ionic 命令行界面，深入介绍 CLI 的功能。
- 第 4 章，Angular 和 TypeScript 基础，介绍 Angular 和 TypeScript 基础知识。
- 第 5 章，Apache Cordova 基础，介绍 Apache Cordova 以及为什么它会被纳入 Ionic 框架的一个组成部分。

- 第 6 章, 理解 Ionic, 简单介绍一个 Ionic 页面由什么构成。
- 第 7 章, 编写我们的 Ionic2Do App, 创建一个基于 Firebase 的 to-do 应用程序。
- 第 8 章, 创建一个基于 Tab 的 App, 使用 Tab 模板编写一个国家公园浏览程序, 集成 Google 地图。
- 第 9 章, 创建一个天气应用, 使用 Forecast.io 天气 API 和谷歌地理编码 API, 编写一个带侧边栏菜单的应用程序。
- 第 10 章, 调试并测试你的 Ionic 应用, 介绍通过一些常用工具解决开发中出现的问题。
- 第 11 章, 部署你的应用程序, 一步一步教你如何将应用提交到应用商店。
- 第 12 章, 探索 Ionic Cloud, 探讨由 Ionic 平台提供的其他服务。
- 第 13 章, 渐进式 Web 应用, 讨论如何用 Ionic 作为一个基础的渐进式 Web App。
- 第 14 章, 终章, 介绍其他 Ionic 组件并扼要介绍了一些其他资源。
- 附录 A, 从 Ionic 1 升级到 Ionic 2, 介绍二者间的重要改变。
- 附录 B, 理解 Config.xml 文件, 介绍和应用程序编译过程有关的各种属性配置。
- 附录 C, Ionic 组件库, 罗列每个可用的 Ionic 组件及一般用法。

所有的代码放在 GitHub 上, 如果你不想手敲书中的示例代码, 或者你想看最新的示例代码, 请访问代码库并下载它的源代码。

如果你做过 Ionic 1 的开发, 可以跳过第 1~3 章。如果你熟悉 TypeScript 和 Angular 2, 请略过第 4 章。如果你会使用 Apache Cordova 或 PhoneGap, 请略过第 5 章。

在线资源

下列资源对于每个 Ionic 开发者来说都是一个很好的开始, 它们总是手可及的:

- 官方 Ionic API 文档 (<http://ionicframework.com/docs/>) 。
- 官方 Angular 2 文档 (<https://angular.io/docs/t/latest/>) 。
- 官方 Apache Cordova 文档 (<https://cordova.apache.org/docs/en/latest/>) 。
- Ionic 全球 Slack 频道 (<http://ionicworldwide.herokuapp.com/>) 。

本书排版约定

本书的排版方式约定如下：

斜体 (*Italic*)

表示新出现的术语、URL、email、文件名及扩展名。

等宽字体 (`Constant Width`)

在代码清单中使用，或者在段落中用于表示程序中的对象，例如变量名、函数名、数据库、数据类型、环境变量、语句和关键字。

粗体等宽字体 (**`Constant Width bold`**)

表示命令或需要用户输入的其他文本。

斜体等宽字体 (*`Constant Width Italic`*)

表示文本应该由用户自己提供的内容替换，或者内容应根据上下文改变。



表示提示或建议。



表示一般的注意事项。