

# 基于Ionic和Cordova的跨平台移动APP的研究与应用

朱凯南,李艳平,申闫春,魏邓航,余越

(北京信息科技大学 计算机学院,北京 100101)

摘要: Ionic是一个专注于用WEB开发技术,基于HTML5创建类似于手机平台原生应用的开发框架。该文讨论了如何基于Ionic和Cordova(PhoneGap)开发跨平台移动端HybridApp。使用web技术开发App,要应对很多移动端原生功能调用的问题,Cordova的插件提供了丰富的原生接口,该文中也介绍了部分插件的应用。

关键词: Ionic;Cordova;HTML5;跨平台;HybridApp

中图分类号: TP311 文献标识码: A 文章编号: 1009-3044(2016)01-0119-03

The Research of Multi-platform Apps Based On Ionic And Cordova(PhoneGap)

ZHU Kai-nan, LI Yan-ping, SHEN Yan-chun, WEI Deng-hang, YU Yue

(School of Computer Science of Beijing Information Science & Technology University, Beijing 100101, China)

Abstract: Free and open source, Ionic offers a library of mobile-optimized HTML, CSS and JS components, gestures, and tools for building highly interactive apps. In this paper, how to build a multi-platform hybrid app based on Ionic and Cordova is discussed. Developing apps with web model will come to some problems on how to use the native functions provided by the mobilephone, Cordova plugins supplied rich APIs to solve these problems, this paper relates to some useful plugins.

Key words: Ionic;Cordova;HTML5;Multi-platform;HybridApp

## 1 概述

### 1.1 背景

智能手机已经成为了人们生活中不可或缺的工具,移动互联网也越来越成为人们生活、学习、工作和娱乐的载体,而随着手机硬件的不断升级,WIFI的广泛覆盖,以及4G网络的发展,使得多种多样功能丰富的应用在移动端的实现成为潮流。但是市场上存在着多操作系统,多型号的移动智能终端,单就安卓一家来看,开发者就几乎要应对碎片级的屏幕和系统差异,这大大提高可移动应用开发的成本。

在本文中我们讨论如何用Ionic, AngularJs等web技术和Cordova打包封装来应对跨平台的移动应用开发问题。

### 1.2 混合模式应用概念阐述

混合模式(hybrid)应用是相对于Native App和Web App而言的。Native App指的是依托于手机本地操作系统,使用原生程序编写的应用程序。Web App指的是采用HTML5语言写出,生存于浏览器中的网页应用。

而Hybrid App则是这两种模式混合开发出的应用程序,也需要下载安装,但访问的内容是基于Web的,拥有Web App跨平台和终端的优点,同时又看起来像一个Native App,能够使用提供各种系统级别的服务,有良好的用户体验。

## 2 基于Ionic和Cordova的跨平台应用创建与开发

### 2.1 应用设计选型

移动端的前端展示,基于Ionic的UI框架和UI组件,混合HTML和CSS来完成,可以应对Android和IOS的跨平台和屏幕适配问题。业务逻辑采用AngularJs和原生Javascript来编写,AngularJs遵循软件工程的MVC模式,为web应用像传统服务器端一样编程提供了解决方案。

涉及移动设备的原生接口的调用,采用Cordova的插件和API接口,最终用Cordova打包发布适用于不同平台的应用。

如果应用涉及到服务器端的支持,采取轻量的数据格式json与服务器端进行数据交互,而对服务器端程序编写的语言和模式几乎没有限制。考虑到移动端和服务器端交互的通信和功能的性能,可以采用RESTful架构。

### 2.2 Cordova及其运行机制简述

Cordova是一款开放源代码的App开发框架,旨在让开发者使用HTML、Javascript、CSS等Web APIs开发跨平台的移动平台应用程序,其原名为PhoneGap,Adobe收购Nitobi公司后,PhoneGap商标保留,代码贡献给了Apache基金会,而Apache将其命名为Apache Callback,其后发布新版本时,定名为Apache Cordova。

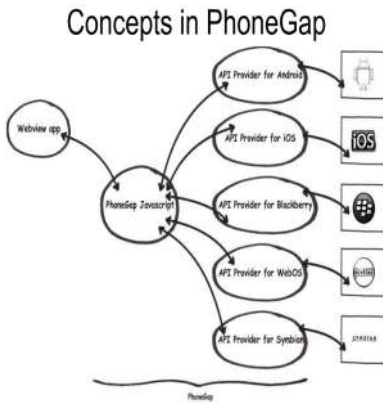


图1 PhoneGap与设备本地API通信图

Cordova架构拥有强大的跨平台访问设备能力,但是其工作原理并不神秘,iPhone和Android平台共同点是都有内置的WebView组件,其具备两个特性:WebView组件实质是移动设备的内置浏览器,WebView这个内置浏览器特性是Web能被打包成本地客户端的基础,可方便的用HTML5、CSS3页面布局,这是移动Web技术的优势相对于原生开发;WebView提供Web和设备本地API双向通信的能力。

Cordova针对不同平台的WebView做了扩展和封装,使WebView这个组件变成可访问设备本地API的强大浏览器,所以开发人员在Cordova框架下可通过JavaScript访问设备本地API。



图2 PhoneGap架构图

## 2.3 Ionic 简述

### 2.3.1 Ionic CSS和Ionic JS

Ionic 是目前最有潜力的一款HTML5手机应用开发框架。通过SASS构建应用程序,它提供了很多UI组件来帮助开发者开发强大的应用。它使用JavaScript MVVM框架和AngularJS来增强应用。提供数据的双向绑定,使用它成为Web和移动开发者的共同选择。Ionic是一个专注于用WEB开发技术,基于HTML5创建类似于手机平台原生应用的一个开发框架。Ionic框架的目的是从web的角度开发手机应用,基于PhoneGap的编译平台,可以实现编译成各个平台的应用程序。

Ionic提供了丰富的CSS库,包括导航栏,页脚,按钮,列表,菜单,toggle,tab切换和网格布局等,Ionic还提供了丰富的颜色样式和icon库,这些样式和图标看起来就像原生的应用一样,并且适配了Android和IOS。Ionic既是一个CSS框架也是一个Javascript UI库。许多组件需要Javascript才能产生神奇的效果,尽管通常组件不需要编码,通过框架扩展可以很容易地使

用,比如AngularIonic扩展。Ionic提供了例如Slide Box(滑动框)、Modal(模型)、Action Sheet(操作表)、Popup(弹出窗口)、Loading(加载)、Platform(平台)、Gesture(手势)、Backdrop(背景)、Utility(工具)、Keyboard(键盘)的UI组件。

### 2.3.2 Ionic 开发环境

#### 1) 安装Ionic和Cordova

首先需要安装Node.js。其次,安装最新版本的cordova和ionic command-line tools。通过npm安装:

```
$ npm install -g cordova ionic
```

#### 2) 通过Ionic创建一个项目

使用Ionic官方提供的现成的应用程序模板,或一个空白的项目创建一个Ionic应用。(ionic start <project-name> <optional-template>) 可选模板为sidemenu 侧滑、tabs 底部tab切换、blank 空白)。本文以tabs模板为例:

```
$ ionic start myApp tabs
```

#### 3) 开发和调试程序

创建好项目之后,就可以用像开发一个Web App一样开发应用了,可以随时在浏览器中打开应用页面来实时调试,或者使用ionic serve --lab命令来展现Ionic项目结果,这个命令会同时以两个面板显示iOS和Android上的运行效果。

#### 4) 打包和运行Ionic项目

Ionic集成了Cordova的编译环境,因此可以直接使用Ionic的命令行工具代替Cordova的命令来打包和运行项目。Cordova的编译依赖于原生程式,所以要先安装好相应平台的sdk等。以安卓环境为例:

```
$ cd myApp
```

```
$ ionic platform add android
```

```
$ ionic build android
```

```
$ ionic emulate android
```

## 3 Cordova 插件的应用

Web技术开发的应用很好地解决了跨平台的问题,可是当涉及到移动端原生功能的调用的时候,比如文件传输,媒体播放,推送,弹窗,本地存储,罗盘等等,就需要应用Cordova插件来实现。本文以本地存储SQLite插件和通知推送Local-Notification插件为例简述Cordova插件的应用。

### 3.1 SQLite 插件

SQLite是一种轻量级的嵌入式数据库,首先在应用目录下通过命令行安装SQLite插件:

```
$ cordova plugin add io.litehelpers.cordova.sqlitestorage
```

然后使用PouchDB(一个javascript库)来操作SQLite,下载PouchDB库然后在应用的首页引入:

```
<script src="lib/pouchdb/dist/pouchdb.min.js"></script>
```

接下来就可以使用PouchDB来操作SQLite了,例如新建数据库:

```
new PouchDB([name], [options])
```

删除数据库:

```
db.destroy([options], [callback])
```

创建一个新的document:

```
db.post(doc, [options], [callback])
```

更新一个document:

```
db.put(doc, [docId], [docRev], [options], [callback])
```

查取一个 document:

```
db.get(docId, [options], [callback])
```

删除一个 document:

```
db.remove(doc, [options], [callback])
```

PouchDB 的所有的数据库操作函数都是异步的并且已经使用了 promise,更多的操作可以查询官方 API 文档。

### 3.2 Local-Notification 插件

Local-Notification 插件用来当应用在后台运行时向用户发送消息和通知。在应用目录下使用命令行安装插件:

```
cordova plugin add de.appplant.cordova.plugin.local-notification@0.8.1
```

Local-Notification 插件 iOS, Android 和 Windows 平台, 功能丰富, 可以设置通知的标题, 文字, 声音, 图片, 提醒时间等等, 例如设置单个通知:

```
cordova.plugins.notification.local.schedule({
  id: 1,
  text: "Single Notification",
  sound: isAndroid ? 'file://sound.mp3' : 'file://beep.caf',
  data: { secret: key }
});
```

设置一个延时通知:

```
var now = new Date().getTime(),
    _5_sec_from_now = new Date(now + 5*1000);
```

```
cordova.plugins.notification.local.schedule({
  text: "Delayed Notification",
  at: _5_sec_from_now,
  led: "FF0000",
  sound: null
});
```

设置一个重复通知:

```
cordova.plugins.notification.local.schedule({
  text: "Delayed Notification",
  firstAt: monday,
  every: "day",
  icon: "file://img/logo.png"
```

```
}, callback);
```

完整的功能可以查询 Local-Notification 在 github 上的文档。

### 4 结束语

Ionic 是一个比较新的开源框架, 仍然在不断地更新和升级, 在国内外也已经有了很火热的社区, 在国内会被越来越多的开发者使用。Cordova 作为 Apache 的项目, 前身 PhoneGap 已经得到了很大的认可, 在插件平台上有很多的开发者在贡献高质量的开源的插件。随着 H5 和移动互联网的发展, 基于 Ionic 和 Cordova 来开发跨平台的移动应用是值得尝试的选择。

### 参考文献:

- [1] 张亚飞, 崔巍. PhoneGap 3.0 移动应用开发实战[M]. 北京: 中国铁道出版社, 2015.
- [2] Leonard Richardson, Mike Amundsen. RESTful Web APIs 中文版[M]. 赵震一, 李哲, 译. 北京: 电子工业出版社, 2014.
- [3] Zakas Nicholas C. JavaScript 高级程序设计[M]. 李宋峰, 曹力, 译. 北京: 人民邮电出版社, 2012.
- [4] David Flanagan. JavaScript 权威指南[M]. 淘宝前端团队, 译. 北京: 机械工业出版社, 2012.
- [5] 郭玉江. 基于 HTML5 的跨平台移动内容管理系统的设计与实现[D]. 杭州: 杭州电子科技大学, 2015.
- [6] 冯永亮. HTML5 本地数据存储技术研究[J]. 西安文理学院学报: 自然科学版, 2014(3): 66-69.
- [7] Green B, Seshadri S. 用 AngularJS 开发下一代 Web 应用[M]. 大漠穷秋, 译. 北京: 电子工业出版社, 2013.
- [8] 行者无疆. 最佳(也许)实践: 开始用 Cordova + Ionic + AngularJS 开发你的 App [EB/OL]. [2015]. <http://segmentfault.com/a/1190000002954226>.
- [9] Ionic. Ionic Documentation[EB/OL]. [2015]. <http://ionicframework.com/docs/>.
- [10] www.ionic.wang. Ionic css 文档中文版[EB/OL]. [2015]. [http://www.ionic.wang/css\\_doc-index.html](http://www.ionic.wang/css_doc-index.html).
- [11] www.ionic.wang. Ionic Javascript 文档中文版[EB/OL]. [2015]. [http://www.ionic.wang/js\\_doc-index.html](http://www.ionic.wang/js_doc-index.html).
- [12] pouchdb. API Reference[EB/OL]. [2015]. <http://pouchdb.com/api.html>.