

# 目录

前言	1.1
APP开发简介	1.2
APP相关项目常见模式	1.2.1
常见APP模式	1.2.2
移动设备类型	1.2.3
APP开发总结	1.3
消息推送	1.3.1
服务商	1.3.1.1
极光推送JPush	1.3.1.1.1
开发总结	1.3.1.2
iOS消息推送	1.3.1.2.1
Android消息推送	1.3.1.2.2
调试移动端Webview/H5页面	1.3.2
调试iOS的Webview	1.3.2.1
调试Android的Webview	1.3.2.2
通话录音	1.3.3
iOS通话录音	1.3.3.1
Android通话录音	1.3.3.2
APP托管总结	1.4
APP托管平台	1.4.1
fir.im	1.4.1.1
蒲公英pgyer	1.4.1.2
APP上架总结	1.5
iOS上架	1.5.1
内测：AdHoc	1.5.1.1
公开市场：AppStore	1.5.1.2
内部发布：企业版	1.5.1.3
OTA版	1.5.1.3.1
Android上架	1.5.2
APP涉及后期运维总结	1.6
数据统计	1.6.1
TalkingData	1.6.1.1
友盟UMeng	1.6.1.2
Bugly	1.6.1.3
崩溃日志收集	1.6.2
Bugly	1.6.2.1
应用内问题反馈	1.6.3

---

蒲公英pgyer	1.6.3.1
iOS开发总结	1.7
开发语言	1.7.1
Objective-C	1.7.1.1
Swift	1.7.1.2
开发工具	1.7.2
XCode	1.7.2.1
AppCode	1.7.2.2
苹果开发者账号	1.7.3
好用的库和工具	1.7.4
Android开发总结	1.8
开发语言	1.8.1
Java	1.8.1.1
Kotlin	1.8.1.2
开发工具	1.8.2
Eclipse+ADT	1.8.2.1
Android Studio	1.8.2.2
好用的库和工具	1.8.3
附录	1.9
参考资料	1.9.1

---

# 移动端APP开发总结

最新版本： v0.2

## 简介

之前折腾过少量的Android开发和一段时间的iOS开发，也了解过混合App的开发，也涉及过基于H5页面加原生壳的APP开发等等，所以此处去整理出这些关于移动端APP开发的各种方面的技术和经验心得，供参考。

## 源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

### Gitook源码

- [crifan/mobiel\\_app\\_summary: 移动端APP开发总结](#)

### 在线浏览

- [移动端APP开发总结 book.crifan.com](#)
- [移动端APP开发总结 crifan.github.io](#)

### 离线下载阅读

- [移动端APP开发总结 PDF](#)
- [移动端APP开发总结 ePUB](#)
- [移动端APP开发总结 Mobi](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2018-02-06 23:23:41

# APP开发简介

在详细介绍移动端APP开发的各种细节之前，先对移动端APP的基本情况做个简介。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-05 22:41:33

## APP相关项目常见模式

APP相关的项目的开发，目前常见有如下几种方案/模式：

### 纯Web模式

PC端：Web页面 手机端：（移动端）Web页面，页面适配手机/Pad等屏幕大小

方案：

- Web页面
  - JS框架：
    - jQuery
    - Angular JS
    - Vue.js
    - React JS
  - UI框架
    - Bootstrap

### PC端Web+App混合模式

PC端：一套Web页面，适配PC端 移动端：一套Web页面，适配移动端 + APP端打包一个壳，内置Webview，加载显示web页面

方案：

- Web页面：
  - JS框架：
    - jQuery
    - Angular JS
    - Vue.js
    - React JS
- APP端
  - iOS
    - 开发工具：Android Studio
      - 只是开发个壳，嵌入chrome的webView，用webView加载web页面
  - Android
    - 开发工具：(Mac) Xcode
      - 只是开发个壳，嵌入safari的UIWebView，用UIWebView加载web页面
        - 另外一种，虽然和Webview+原生的壳不同，但是也属于广义上的混合APP的模式：
        - React Native的混合模式
        - 用React Native写Javascript代码，一套代码
        - -> 生成iOS和Android的原生的app

### PC端Web + App端：原生APP

- Web页面：
  - JS框架：
    - jQuery

- Angular JS
- Vue.js
- React JS
- APP端
  - iOS
    - 开发工具: Android Studio
      - 开发完整的独立的app的完整功能
    - Android
      - 开发工具: (Mac) Xcode
        - 开发完整的独立的app的完整功能

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-05 22:57:47

## 常见APP模式

正如前面的介绍，一般项目涉及到的APP模式，可以简化立即为有三种：

- 纯Web=纯Web页面适配移动端
- Web App=移动端混合APP
- Native App=纯原生移动端APP

而其中的 纯Web的，指的是一套Web页面，完全适配移动端浏览器，所以还是属于基于浏览器的页面，不算传统意义上的移动端的APP，所以不做详细解释。

接下来主要解释：混合式的Web App和纯的Native原生APP。

## Web App对比Native App

详细解释之前，对于典型的Web的APP和原生的Native的APP进行一个简单的对比：

Native App	Web App	说明
偏交互	偏浏览	交互指复杂操作，输入/选择什么的
已稳定	试错中	H5页面用来做低成本验证很好
访问硬件	信息展示	指手机里的各种传感器什么的
核心功能	周边辅助	把工作量多投在刀刃上

## 原生Native

原生的APP，概念上其实很简答，就是指的是：

- iOS平台：典型的指的是，在Mac的XCode下，用纯的之前的Objective-C或最新的Swift的代码开发和调试，最终生成的原生的iOS的ipa文件
- Android：在Mac/Windows系统中，用之前的Eclipse+ADT或最新的Android Studio工具，写之前的Java或最新的Kotlin代码，去开发调试出来的Android原生的apk文件

## 混合APP

关于混合APP开发，往往指的是：用一套框架或技术，最终得到的app。

### H5+原生的壳

其中，用原生的app写个基本的启动界面，加上一个WebView，然后用webview去加载所有的app的内容，即一堆的HTML+CSS+JS，从而实现app的效果。

所以，就分成两部分：

- H5页面=Webview页面
- 原生的壳=原生代码(iOS/Android)去写对应的app客户端
  - 可能还包括一些其他相关的基本功能：
    - 启动页面
    - 自动保存密码

- 用于实现自动登录
- 消息推送
  - 纯Web页面无法（很难）做到消息推送，只能靠原生

如此，H5+原生的壳，就实现了一个移动端的APP，如果效果做的还可以，基本上可以以假乱真，看起来和原生的APP没多大区别。

对于这类移动端的Web页面的设计，也有专门的逻辑和注意事项，比如：

[Mobile Web Development - Web developer guides | MDN](#)

## 混合APP框架

另外一种实现一般所谓的混合APP的做法就是：使用某个混合APP框架去实现

而混合式APP框架，又可以分为两类：

- 传统混合APP框架
- 新出的一些特殊的框架

下面分别来介绍。

### 传统混合APP框架

传统的框架，有很多，其中一些比较出名的有：

#### **cordova(旧称PhoneGap)**

之前叫PhoneGap，后被Adobe收购了改名为cordova。

主页：[Apache Cordova](#)

#### **ionic**

还有一个混合APP框架 `ionic`，好像用的也不少

主页：[Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular](#)

## 新出现的基于JavaScript的框架

后来又出现一些，从广义上也可以被称为混合式APP框架的：

#### **React Native**

`React Native = RN`

Facebook出的，可以用一套 `React Native` 框架，写JS代码，最终生成iOS和Android两端的App。

性能总体上比纯Web要好。

主页：[React Native · A framework for building native apps using React](#)

#### **weex**

阿里的出的，目标是支持一套代码，生成iOS、Android和Web端的程序。

常常被拿来和RN对比。

目前流行度好像一般。

主页：[Weex](#)

## HBuilder

还有个叫做 HBuilder，好像不是很流行，但是之前简单尝试过，没具体折腾。

好像也可以生成APP，但是感觉是一个官网的原生的APP壳，加上内部的Webview的效果。

也算作其中一个框架吧，虽然可能是从功能上来说是比较弱的。

主页：[DCloud - HBuilder、5+、mui、流应用、HTML5专家](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2018-02-05 23:23:51

# 移动设备类型

## 按操作系统分

### 苹果的iOS

苹果的移动端的设备，典型都有iPhone，iPad等，所用的操作系统都是iOS。

iOS的设备类型，相对较固定，数量有限。

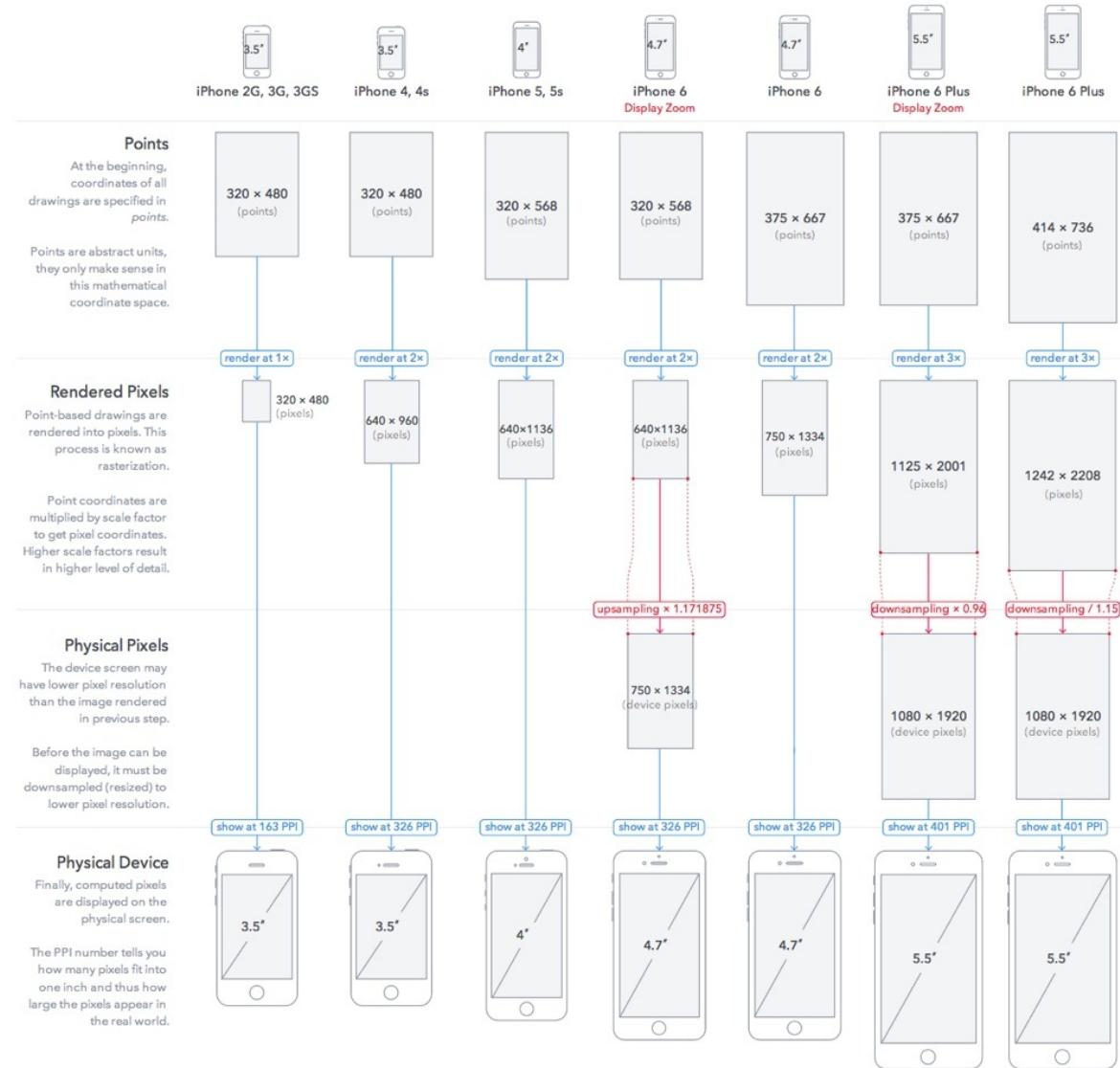
### iOS的设备类型和尺寸

下面总结一下，不同的iPhone设备的屏幕相关的参数：

机型	屏幕宽高/单位点	屏幕模式	屏幕对角线长度
iPhone 3GS	320x480	1x	3.5英寸
iPhone 4	320x480	2x	3.5英寸
iPhone 5	320x568	2x	4英寸
iPhone 6	375x667	2x	4.7英寸
iPhone 6 Plus	414x736	3x	5.5英寸

对应的屏幕类型分辨率和像素的换算关系为：

## The Ultimate Guide To iPhone Resolutions



## Google的Android

Google只是研发了Android的移动设备的操作系统，而普通用户买到的手机，都是各个厂商，比如小米、华为、联想等，加上硬件和研发最终设计出来的各自品牌的手机。

也因此，Android的设备类型，相对iOS的有限的设备来说，就显得多而杂，有高端有低端，有大屏有小屏，而且各自等屏幕尺寸又不一定完全相同。

## Android的常见设备列表

之前（好像是）从：[支付宝行走支持的机型\\_红米手机1\\_MIUI论坛](#)中，参考整理了如下等Android等设备列表，供参考：

- 小米
  - 小米3C/小米3W
  - 小米4C/小米4W/小米4i/小米4S
  - 小米5/小米5S/小米5SPlus
  - 小米MIX/小米MAX

- 小米Note2/小米Note Pro//小米Note LTE
- 小米平板/小米平板2
- 红米1S
- 红米3S/红米3X
- 红米4/红米4A
- 红米Note1/红米Note3 全网通版
- OPPO
  - A30/A57/A59m/A59s
  - Find7(X9077/X9007/X9000/X9070)
  - N3(N5207/N5209)
  - R3(R7005)
  - R9(R9km/R9m/R9s/R9t/R9tm/X9009)
  - R9 Plus(R9 PlusmA/R9 Plust A/R9 Plutm A)
  - R9s(R9st/R9sk)
  - R9s Plus
  - VIVOX3 V
  - X5 M/X5 M L/X5 Max L/X5 Max+/X5 Pro L/X5 Pro V
  - X6 D/X6 L/X6 Plus D/X6 Plus L/X6 S/X6 SA/X6 S Plus/X6 S Plus A/X6 S Plus D
  - X7/X7 L/X7 10F/X7 10L/X7 Plus/X7 Plus L
  - X9
  - Xplay3S(X520F)/Xplay5A/Xplay5S/Xplay6
  - Y55A/Y67
- 魅族
  - M5 Note/MX4/MX4 Pro/MX4 YUNOS版/MX5/MX6
  - Pro5/Pro6/Pro6 Plus/Pro6s
  - 魅蓝E/魅蓝Metal/魅蓝M3 Max/魅蓝Note/魅蓝Note2 公开版 移动定制版 联通定制版/魅蓝Note3/魅蓝U20
- 华为
  - 手机
    - G9青春版(VNS-DL00/ VNS-AL00/VNS-TL00)/G9 Plus(MLA-TL00/MLA-UL00/ MLA-TL10)
    - Nova(CAZ-AL10/ CAZ-TL10)
    - Mate 7(CL00/UL00/TL00/TL10/L09)
    - Mate 8(AL10/CL00/DL00/TL00/eH880/NXT-L29)
    - Mate 9(MHA-AL00)/Mate 9 Pro(LON-AL00)
    - Mate S(CRR-CL00/CRR-UL00/CRR-TL00/CRR-CL20/CRR-UL20)
    - P8(GRA-CL00/GRA-CL10/GRA-TL00/GRA-UL00/GRA-UL10)/P8 Max
    - P9(EVA-AL00/EVA-AL10/EVA-CL00/EVA-DL00/EVA-TL00/EVA-L19)
    - P9 Plus(VIE-L29/VIE-AL10)
    - 麦芒5(MLA-AL00/MLA-AL10)
    - 荣耀6Plus(PE-UL00/PE-TL20/PE-TL10/PE-TL00M/PE-CL00)
    - 荣耀6(H60-L02/H60-L11/H60-L01/H60-L03/H60-L12/H60-L21)
    - 荣耀7(PLK-CL00/PLK-AL10/PLK-TL01H/PLK-UL00/PLK-TL00)
    - 荣耀8(FRD-AL00/FRD-AL10/FRD-DL00)/荣耀 Note8(EDI-AL10)
    - 荣耀X2(GEM-703L/GEM-703LT)
    - 荣耀V8(KNT-AL10/KNT-AL20/KNT-TL10/KNT-UL10)
    - 荣耀畅玩5A(CAM-AL00)
    - 荣耀畅玩5C(NEM-AL10/NEM-TL00H/NEM-TL00/NEM-UL10)
    - 荣耀畅玩6X(BLN-AL10/BLN-TL10)
  - 平板
    - (M2-A01L/M2-A01w/M3 BTV-DL09/M3 BTV-W09)
- 三星
  - A5(A5100/A5108)/A7(A7100/A7108)/A9(A9000/A9100)/Alpha(G8508S)

- C5(C5000)/C7(C7000)/C9 Pro(C9000)
- J(SC-02F)
- Note 3(N7505/N7509V/N9007/N900J/N900K/N900L/N900P/N900V/SC-01F/N900/N9002/N9005/N9006/N9008/N9008S/N9008V/N9009/N 900S)
- Note 3 Lite(N7506V/N7508V)
- Note 4(N910C/N910F/N910G/N910K/N910L/N910P/N910S/N910U/N9100/N9106W/N9108V/N9109W)
- Note 5(N9208/N920C/ N920K/N920L/N920P/N920S/N920V/N9200)
- Note 7(N9300)
- Note Edge(N9150/N915S)
- S4(I9500/I9502/I9505/I9507/I9507V/I9508/I9508V/L720/SCH-I959)
- S4 LTE-A(E330S)
- S5(G9006V/G9006W/G9008V/G9008W/G9009D/G9009W/G900F/G900H/G900I/G900J/G900K/G900L/G900P/G900V/G906K/SCL23)
- S5 LTE-A(G906L/G906S)
- S6(G9200)G9208/G9209/G920F/G920k/G920L/G920P/G920S/G920T/G920V/SC-05G)
- S6 Edge(G920I/G9250/G9250L/G925A/G925F/G925I/G925K/G925P/G925S/G925T/G925V/SCV31)
- S6 Edge+(G9280/G9287/G928A/G928F/G928K/G928P/G928S/G928T/G928V)
- S7(G9300/G9308/G930F/G930K/G930L/G930S/G930T/G930U/G930V)
- S7 Edge(G9350/G935F/G935K/G935L/G935S/G935T/G935u/G935V)
- W2015/W2016
- Galaxy 旗舰III(G9198)
- 乐视
  - 乐 Pro3(X720)
  - 乐1 Pro (X800/ X800+)
  - 乐2(X520/X528/X620/X621)/乐2 Pro(X625)
  - 乐Max(X900/X900+)/乐Max 2(X820/X822)/乐Max Pro(X910)
- 一加
  - 一加手机1 (A0001/A1001)
  - 一加手机2(A2001/One A2001)
  - 一加手机3(OnePlus A3000)
  - 一加手机3T (A3010)
  - 一加手机X (ONE E1001/ONE E1000)
- 360
  - N4(1503-A01/503-M02)/N4A(1603-A03)/N4S(1505-A01/1505-A02)
  - Q4(1509-A00)/Q4青春版(1515-A01)
- 努比亚
  - nubia N1(NX541J)
  - nubia X6(NX601J)
  - nubia Z5s(NX503A)
  - nubia Z7 Max(NX505J)/nubia Z7 Mini(NX507J)/nubia Z7(NX506J)
  - nubia Z9 Max(NX510J/NX518J/NX508J)
  - nubia Z11(NX531J)/nubia Z11Max(NX523J\_V1)/nubia Z11Mini(NX529J)/nubia Z11Minis(NX549J)/nubia z11 Max\*(NX535J)
- 中兴
  - Axon Mini(B2015)/Axon天机(A2015)/Axon天机7(A2017)/Axon天机 Max(C2016/C2017)
  - Blade(BA910/BA910T)/Blade A1(C880U/C880A)/Blade A2 Plus(BV0730)
  - Geek II/Grand S II LTE
  - V5(N9180/X9180)
  - V7 MAX(BV0710/BV0710T)
  - 青漾2S(G718C)
  - 小鲜3(C880S/C880D)/小鲜4(BV0701)

- 星星1号(S2002)/星星2号(S2005/S2010)
  - 远航3(Q529C/Q529T)
- 联想
  - Z1(全网通版)/Z2(Z2131)/Z2 Pro(Z2121)
  - S860e(电信版)
  - 乐檬X3(X3c70/K51c78)/乐檬X3青春版(K52t38)/乐檬K5 Note( K52e78)
  - VIBE P2(P2c72)/VIBE X2(X2-TO)/VIBE Z(K910)/VIBE Z2 Pro(K920/X2Pt5)
- 锤子
  - 坚果/坚果Pro/坚果Pro 2
  - M1/M1 L(SM919)
  - T1(SM701/SM705)/T2(SM801)
- 酷派
  - 酷派C106/C106-9/C106-6/C106-8/派C107-9
  - 锋尚MAX(A8-831/A8-930/A8-932/A8-931N)
  - 大观铂顿 V1-C电信版
- 金立
  - 金立
    - E8移动联通版(GN9008)
    - GN715移动联通版
    - M6(GN8003)/M6 Plus(GN8002/GN8002S)
    - S5.1移动版(GN9005)
    - S6 Pro(GN9012)
    - S7(GN9006)/S7 Mini(GN9007)
    - S8 全网通(GN9011)
    - 大金钢全网通(GN5003)
  - IUNI
    - IUNI N1
    - IUNI i1
    - IUNI U2
    - IUNI U3
- 索尼
  - Xperia C3(S55t/S55u)
  - Xperia T2 Ultra(XM50h/Xm50t)
  - Xperia X (F5122/F5321)
  - Xperia XP(F8132)
  - Xperia XZ(F8332)
  - Xperia Z Ultra(XL39h)
  - Xperia Z1(C6902/C6903/D5503/L39h)/Xperia Z1 mini (M51w)
  - Xperia Z2(L50t/L50u/D6503)
  - Xperia Z3(D6633/D6683/D6708/D6603/D6653/D5803/D5833/L55u/L55t/SO-01G/SO-02G/SOL26)
  - Xperia Z3+ dual (E6553/E6533)
  - Xperia Z5(E6653/E6683)/Xperia Z5 Compact(E5823)/Xperia Z5 Premium(E6883/E6853)
- 摩托罗拉
  - Moto Z(XT1650-05)/Moto Z Play (XT1635-03)
  - Moto M(XT1662)
  - Moto X(XT1060)
  - LGAKA(H778)
  - G Flex(D958/LS996)
  - G Pro 2 (LG-D838)
  - G2(D802/L22/LS980/VS980)
  - G3(D855/D857/D858/D858HK/D859/F400L/F460/F460K/F460S/L24/LS990/VS985)

- G4(F500L/H815/H818/H819)
- G5(F700L/H850/H860/H868/LS992)/G5 SE(H848)
- V10(F600L/F600S/H901/H961N/H968/VS990)
- 谷歌
  - Nexus 5/Nexus 5X/Nexus 6/Nexus 6P
  - Pixel/Pixel xl
  - HTC10(M10h/M10u)
  - Desire(D830u/D10w)
  - One A9/One A9移动联通版
  - One E9(E9t/E9w)/One E9+(E9pw)
  - One M8si
  - One M9(M9u/M9w/m9e)/One M9+(M9pt/M9pw)
  - One ME(M9et/(M9ew)
  - One X9u
- 中国移动
  - A2(M636/双4G)
  - M812C
  - N2(M836)
- 华硕
  - ZenFone2(ASUS\_Z00ADB/ASUS\_Z00AD/ASUS\_Z00ADA)
  - ZenFone Zoom(ASUS\_Z00XSB)
- 海信
  - A1/F3 Pro(F31)/K8(H910)
  - 小岛(E20T )/金盾(E76)
- Gigaset金阶
  - Gigaset ME(GS55-6)
  - Gigaset ME Pro (GS57-6)
- 8848
  - M3
- IVVI
  - IVVI i3(i3-01/SS2-01)
- AGM
  - X1
- PPTV
  - PPTV Mipo M1(PP5600)
- SUGAR
  - SUGAR F7
- TCL
  - 580/750/950
- YotaPhone
  - YotaPhone2 (YD201/YD206)
- 泛泰
  - A890S/A910S
- 飞利浦
  - 杰智I999
- 富可视
  - M2/蓝鲸S1
- 格力
  - 格力手机2代 (G0215D)
- 黑莓
  - STV100-1/STV100-3

- 亚马逊
  - Fire Phone(SD4930UR)

## 按设备尺寸分

**相对大屏的平板Pad**

**想对小屏的手机Phone**

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-06 23:21:17

# APP开发总结

接下来介绍，对于移动端，包括iOS端和Android端，相对来说算是通用的开发相关的知识。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-06 21:19:01

## 消息推送

iOS和Android端，都会涉及到，消息推送。

典型的场景有：

系统发送一个通知到某个用户，需要对应消息提醒，且希望在App没有运行的情况下，就能收到iOS和Android的系统的消息推送

比如，iOS手机中收到一条消息推送：



对于这类场景的消息推送，一般也叫做 离线消息推送。

下面来详细解释消息推送方面的知识。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2018-02-06 23:13:27



## 消息推送服务商

能否实现消息推送的功能的话，一般从功能上，至少包括两部分：

- 消息推送的服务器端：实现消息队列的管理，选择合适的消息发送的时机，且要考虑到APP端可能离线，以及消息重发等等内部复杂逻辑判断。
  - 一般来说，如果不是有特殊需要，往往都是采用使用第三方的消息推送的服务
  - 否则完全自己实现消息推送的后台服务，还是需要一定端技术成本和服务器采购成本的
    - 尤其是把消息推送做到让移动端APP方便调用，且消息推送的效果好，那需要更大的成本
- 消息推送的APP端：
  - 在iOS和Android端，利用系统提供的消息推送端方面的接口，实现消息的接收和处理

对于第三方面端消息推送服务，做的比较好，主要是：极光推送 = JPush，下面就来详细介绍一下极光推送。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-06 21:39:30

# 极光推送JPush

极光推送 = JPush

主页: 首页 - 极光 | 数据改变世界

## 极光推送的价格

默认是有免费的服务使用的。限制是：所有的使用免费的极光推送的APP去共享使用20万/秒的服务。

- -> 换句话说：如果你的APP的消息推送正好赶上其他所有的免费的APP使用极光的服务的高峰期，则消息推送可能会稍微延迟一段时间。
  - 其实一般情况下，多数情况下，延迟也还是很低的，总体的感觉是80%以上的时间，服务器端消息推送后，APP端收到消息的延迟，都是在1秒~5秒，足够一般的APP使用了，不会觉得多慢的
- -> 换另外一句话说：如果真的是你的APP对于消息推送的推送的及时率有很高的要求，那么就应该去购买收费的极光推送的服务了
  - 收费的激光服务的价格：最少每月也要2000多元
    - 所以收费服务，对于小公司，还是挺贵的
    - 所以对于一般的APP来说，如前面所说的极光的免费的消息推送的服务，也就够用了，不需要额外花钱买收费的

另外，对于APP用户规模大的，需要购买收费的极光的服务的，附上价格表，供参考：

The screenshot shows a PDF document titled "极光推送付费服务报价表" (JPush Paid Service Price List). The table details seven service tiers based on user scale, with columns for version type, user scale, peak online users, monthly fee, API frequency, and additional cost per 10,000 users.

版本类型	APP用户规模	在线峰值用户	月费 (元)	Push API 频率 (次/分钟)	增加1万在线用户费用增加 (元)
初级版	10万-20万	2万	2,499	12,00	1000
基本版	50万-100万	10万	9,999	48,00	1000
中级版	100万-250万	30万	25,999	120,00	750
专业版	250万-500万	50万	35,999	240,00	750
高级版	500万-1000万	100万	49,999	480,00	500
中型企业版	1000万-2000万	200万	89,999	1200,00	500
大型企业版	2000万-5000万	500万	199,999	4800,00	500

**计费原则:** 每月根据App当月在线用户峰值加超出部分作为当月扣费标准，当此费用超过下一版本月费时系统自动按下一版本费用扣费。如16万月在线用户峰值当月实际付费为  $9999+6000 = 15999$  元；27万月在线用户峰值实际付费为  $9999+17000 = 26999$ ，则当月付费按下一版本25999计费。

**注:** 日在线用户数：一天之内登录JPush服务器的独立用户数  
月在线用户数峰值：自然月内“日在线用户数”最高的一天的数值

此付费服务报价表解释权归极光推送所有

## 极光推送消息基本知识

## iOS的证书

iOS的消息推送，支持开发环境和测试环境。

需要分别去上传对应的证书才可以的。

The screenshot shows the Jiandao application settings interface. At the top, there are tabs for Push, Statistics, IM, and Application Settings. The current tab is Application Settings. In the main area, there's a section for the app 'com.daryun.jiandao' with a 'Download Android Example' button. Below this is the iOS configuration section:

iOS	
Bundle ID	com.daryun.jiandao
APNS推送环境	生产环境
APNs证书文件	开发环境: 已验证 生产环境: 已验证
证书有效期至	开发环境: 2017-03-10 14:46:49 生产环境: 2017-05-08 20:22:53

Below the iOS section is the WinPhone configuration section:

WinPhone	
启用 WinPhone	否

At the bottom right is a blue '修改应用' (Edit Application) button.

## 标签tag和别名Alias

可以通过tag或alias，去实现，给某类用户设置同一个tag或alias，然后消息推送的时候，给同一类的用户批量推送相同的消息。

比如，牛只管理app中，对于牛只的管理员，当发生异常信息，比如牛只设备掉落时，给所有的同一组的管理员，都发送预警消息。

另外一个典型的使用方式是：

比如对于一个APP，内部有一套自己的用户系统，其中每个用户有自己的userid，比如类似于UUID的这种： user-39ee1299-4e29-43cf-904f-d8826ce1b899

而如果移动端想要实现接收到服务器推送不同的消息给每个用户，其中一种实现方式就是：

每个用户都设置一个，借用userid的独一无二的别名，比如：

user-39ee1299-4e29-43cf-904f-d8826ce1b899

后续服务器端给每个用户推送消息时，就直接给对应别名去推送消息，对应的用户即可收到对应的消息推送了。

其中几点需要说明的是：

- APP中用户登录后，APP初始化时（在init初始化JPush的API后）去注册register对应的自己用户ID对应的别名alias

- 符合正常的逻辑：用户登录后，应该能收到消息推送
- APP中用户退出登录时，去注销对应的JPush的对应的别名
  - 这样后续该用户就不会收到消息提醒了
  - 符合正常的逻辑：用户注销后，不应该继续收到消息推送

## iOS的JPush相关示例代码

附上之前的iOS的app：简道中的JPush的消息推送的iOS端的相关代码，供参考：

APP初始化部分的JPush相关配置：

文件：`AppDelegate.swift`

代码：

```
//for com.daryun.jiandao
//let JpushAppKey:String      = "3a1a8d7c1caa422ee6648da3"
// masterSecret: 75b70670870934f8443e98e5
//for im.jiandao.app
let JpushAppKey:String      = "54ab761c3b551e1cf8dcdb47"
// masterSecret: 6e97a9601552641bbc122f20
let JpushChannel:String     = "iOS-AppStore"
//let JpushIsProduction:Bool = false
let JpushIsProduction:Bool = true

func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    gLog.debug("didFinishLaunchingWithOptions launchOptions=\(launchOptions)")

    //1.register device
    JPUSHService.registerForRemoteNotificationTypes(UIUserNotificationType.Badge.rawValue | UIUserNotificationType.Sound.rawValue | UIUserNotificationType.Alert.rawValue, categories: nil)
    JPUSHService.setupWithOption(launchOptions, appKey: JpushAppKey, channel: JpushChannel, apsForProduction: JpushIsProduction)
    listenRemotePush()

    return true
}

func application(application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken deviceToken: NSData) {
    //2.receive device token for register
    gLog.debug("deviceToken=\(deviceToken)")
    //deviceToken=<77366f0d c4a4f2f4 74fe24de 5db46132 c725ee1a 8e41b1fc a9a511bc 832c0113>
    NSNotificationCenter.defaultCenter().postNotificationName("DidRegisterRemoteNotification", object: deviceToken)
    //3. send device token to push server
    JPUSHService.registerDeviceToken(deviceToken)
}

func application(application: UIApplication, didFailToRegisterForRemoteNotificationsWithError error: NSError) {
    gLog.debug("error=\(error.localizedDescription)")
}

// //for iOS <= 6.0
// func application(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject : AnyObject]) {
//     //4. receive the remote push message from (APNS -> JPUSH) server
//     print("didReceiveRemoteNotification")
//     print("userInfo=\(userInfo)")
//     JPUSHService.handleRemoteNotification(userInfo)
//     NSNotificationCenter.defaultCenter().postNotificationName("AddNotificationCount", object: nil)
// }

//for iOS >= 7.0
func application(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject : AnyObject], fetchCompletionHandler completionHandler: (UIBackgroundFetchResult) -> Void) {
    gLog.debug("userInfo=\(userInfo)")
    JPUSHService.handleRemoteNotification(userInfo)
    NSNotificationCenter.defaultCenter().postNotificationName("AddNotificationCount", object: nil)
    NSNotificationCenter.defaultCenter().postNotificationName("handleRemoteNotification", object: userInfo)
    completionHandler(UIBackgroundFetchResult.NewData)
}
```

```

}

func application(application: UIApplication, didReceiveLocalNotification notification: UILocalNotification) {
    gLog.debug("notification=\\" + notification + "\")

    //notification=<UIConcreteLocalNotification: 0x15f158b60>{fire date = 2016年3月11日 星期五 中国标准时间 16:33:53, ti
    me zone = Asia/Shanghai (GMT+8) offset 28800, repeat interval = 0, repeat count = UILocalNotificationInfiniteRepeatCount,
    next fire date = (null), user info = (null)
    JPUSHService.showLocalNotificationAtFront(notification, identifierKey: nil)

    //
    if let userInfo = notification.userInfo {
    //
        let newMessage = userInfo["newMessage"] as! Message
    //
        debugPrintMsg(newMessage)
    //
    }
}

@available(iOS 7, *)
func application(application: UIApplication, didRegisterUserNotificationSettings notificationSettings: UIUserNotificat
ionSettings) {
    gLog.debug("notificationSettings=\\" + notificationSettings + "\"")
    //notificationSettings=<UIUserNotificationSettings: 0x79070130; types: (UIUserNotificationTypeAlert UIUserNotifica
tionTypeBadge UIUserNotificationTypeSound);>
    /*
        notificationSettings=<UIUserNotificationSettings: 0x12e6529a0; types: (UIUserNotificationTypeAlert UIUserNotificat
ionTypeBadge UIUserNotificationTypeSound);>
        2016-03-11 16:22:05.110 | JPUSH | W - [JPUSHClientController] Not get deviceToken yet. Maybe: your certificate not
configured APNs? or current network is not so good so APNs registration failed? or there is no APNs register code? Pleas
e refer to JPUSH docs.

        2016-03-11 16:22:05.113 | JPUSH | I - [JPUSHSessionController] sis is not on protect
        2016-03-11 16:22:05.116 | JPUSH | I - [JPUSHAddressController] Action - sendSisRequest
    */
}

@available(iOS 7, *)
func application(application: UIApplication, handleActionWithIdentifier identifier: String, forLocalNotification noti
cation: UILocalNotification, completionHandler: () -> Void) {
    gLog.debug("handleActionWithIdentifier=\\" + identifier + "\", forLocalNotification=\\" + notification + "\"")
}

@available(iOS 7, *)
func application(application: UIApplication, handleActionWithIdentifier identifier: String, forRemoteNotification use
rInfo: [NSObject : AnyObject], withResponseInfo responseInfo: [NSObject : AnyObject], completionHandler: () -> Void) {
    gLog.debug("handleActionWithIdentifier=\\" + identifier + "\", forRemoteNotification=\\" + userInfo + "\", withResponseInfo=\\" + respon
seInfo + "\"")
}

func networkDidLogin(notification:NSNotification) {
    gLog.debug("已登陆 networkDidLogin notification=\\" + notification + "\"")

    if let registrationID = JPUSHService.registrationID() {
        gLog.debug("registrationID=\\" + registrationID + "\"")
    }
}

func listenRemotePush(){
    gLog.debug("")
    let defaultCenter:NSNotificationCenter = NSNotificationCenter.defaultCenter()
    defaultCenter.addObserver(self, selector: #selector(AppDelegate.networkDidSetup(_:)), name:kJPFFNetworkDidSetupNoti
fication, object: nil)
    defaultCenter.addObserver(self, selector: #selector(AppDelegate.networkDidClose(_:)), name:kJPFFNetworkDidCloseNoti
fication, object: nil)
    defaultCenter.addObserver(self, selector: #selector(AppDelegate.networkDidRegister(_:)), name:kJPFFNetworkDidRegist
erNotification, object: nil)
    defaultCenter.addObserver(self, selector: #selector(AppDelegate.networkDidLogin(_:)), name:kJPFFNetworkDidLoginNoti
fication, object: nil)
    defaultCenter.addObserver(self, selector: #selector(AppDelegate.networkDidReceiveMessage(_:)), name:kJPFFNetworkDid
ReceiveMessageNotification, object: nil)
    defaultCenter.addObserver(self, selector: #selector(AppDelegate.serviceError(_:)), name:kJPFSERVICEERRORNotificati
on, object: nil)

    defaultCenter.addObserver(self, selector: #selector(AppDelegate.didRegisterRemoteNotification(_:)), name:"DidRegis
terRemoteNotification", object: nil)
}

```

```

        defaultCenter.addObserver(self, selector: #selector(AppDelegate.handleRemoteNotification(_:)), name:"handleRemoteNotification", object: nil)
    }

    func handleRemoteNotification(remoteNotification:NSNotification) {
        gLog.debug("remoteNotification=\(remoteNotification)")
        if let userInfoDict = remoteNotification.object {
            gLog.debug("userInfoDict=\(userInfoDict)")

            if let aps = userInfoDict["aps"] {
                if let badge = aps ["badge"] as? Int {
                    gLog.debug("badge=\(badge)")
                    UIApplication.sharedApplication().applicationIconBadgeNumber = badge
                }
            }

            if UIApplication.sharedApplication().applicationState == UIApplicationState.Active {
                gLog.debug("not handle remote notification for app is active running")
            } else {
                gLog.debug("applicationState=\(UIApplication.sharedApplication().applicationState)")

                SingletonConversationTVC().updateAllConversation()
            }
        }
    }
}

```

文件: LoginViewController.swift

代码:

```

func doAfterLogin(){
    dispatchBackground_async({
        //init message related
        SingletonMainVC().initWebSocket()
    })
}

```

文件: MainViewController.swift

代码:

```

func jpushSetAlias() {
    gLog.debug("")
    var userIdAlias = gCurUserItem.id
    userIdAlias = userIdAlias.replace("user-", to: "usr-")
    userIdAlias = userIdAlias.replace("-", to: "_")
    gLog.debug("userIdAlias=\(userIdAlias)")
    //user-e6882cab-cedf-4335-9b7c-612cd5b4d37d
    //usr_e6882cab_cedf_4335_9b7c_612cd5b4d37d

    jpushSetAlias(userIdAlias)
}

func jpushSetAlias(userIdAlias:String) {
    gLog.debug("userIdAlias=\(userIdAlias)")

    JPUSHService.setAlias(userIdAlias, callbackSelector: #selector(MainViewController.aliasCallBack(tags:alias:)), object: self)
}

func jpushClearAlias() {
    gLog.debug("")
    jpushSetAlias("")
}

func aliasCallBack(resCode:CInt, tags:NSSet, alias:NSString) {
    gLog.debug("resCode=\(resCode), tags=\(self.logSet(tags)), alias=\(alias)")
    //resCode=6004, tags=nil, alias=user-e6882cab-cedf-4335-9b7c-612cd5b4d37d
}

```

```

//resCode=6003, tags=nil, alias=e6882cab-cedf-4335-9b7c-612cd5b4d37d
//resCode=0, tags=nil, alias=usr_e6882cab_cedf_4335_9b7c_612cd5b4d37d
if resCode != 0 {
    let failMsg = "设置JPush推送的别名失败: alias=\(alias)"
    gLog.warning(failMsg)
}

func initWebSocket(){
    dispatchUserInitiated_async({
        gLog.debug("gCurUserItem.wsUrl=\(gCurUserItem.wsUrl)")

        //ws://jiandao.im/message/user-972b6796-cc82-4058-b29f-9007115116b9/918fu0uooq18tunpo9h6grp6gk
        self.webSocket = WebSocket(url: NSURL(string: gCurUserItem.wsUrl)!)
        self.webSocket.delegate = self
        self.webSocket.pongDelegate = self
        self.webSocket.connect()

        gCurUserItem.wsInitiated = true

        MainViewController.wsFirstInit = true

        gLog.debug("websocket \(self.webSocket) has initied and connected")

        self.jpushSetAlias()
    })
}

```

而JPush正常注册后的log日志为：

```

didFinishLaunchingWithOptions
2016-03-12 10:35:48.933  JPUSH  I - [JPUSHService]
----- JPush Log -----
-----JPush SDK Version:2.1.0--build:346-----
-----AppKey:3a1a8d7c1caa422ee5548da3-----
-----
2016-03-12 10:35:48.942  JPUSH  I - [JPUSHClientController] Action - setup
prevStoredUser Optional("15051464654")
prevStotedPassword Optional("111111")
didRegisterForRemoteNotificationsWithDeviceToken
deviceToken 89b266fc 31bb2513 20a35909 03a31e9f c8127fef 4946138c 66df7769 617b1c0c
已注册远程通知 didRegisterRemoteNotification (Function)
deviceTokenStr Optional( 89b266fc 31bb2513 20a35909 03a31e9f c8127fef 4946138c 66df7769 617b1c0c )
didRegisterUserNotificationSettings
notificationSettings UIUserNotificationSettings: 0x14dd5cf60; types: (UIUserNotificationTypeAlert UIUserNotificationTypeB
adge UIUserNotificationTypeSound);
2016-03-12 10:35:49.107  JPUSH  I - [JPUSHSessionController] sis is not on protect
2016-03-12 10:35:49.109  JPUSH  I - [JPUSHAddressController] Action - sendSisRequest
已连接 networkDidSetup notification NSConcreteNotification 0x14dda9c20 {name = kJPUSHNetworkDidSetupNotification}
已登陆 networkDidLogin notification NSConcreteNotification 0x14de7f470 {name = kJPUSHNetworkDidLoginNotification}
registrationID 13165ffa4e0b5fb6ba3
2016-03-12 10:35:50.104  JPUSH  I - [JPUSHDeviceTokenReport] upload device token success

```

## JPush的服务不错

后台管理页面中可以方便测试消息推送

可以方便的去测试消息发送

支持对于：

- iOS
  - 生产环境
  - 测试环境
- Android

还可以指定发送对象

- 具体的某个设备
- 某个tag
- 某个alias
- 广播所有设备

等等。

截图供参考：

The screenshot shows a mobile application interface. At the top left is a white rectangular box labeled "测试ios推送1". To its right is a smartphone icon representing an Android device, with the word "Android" above it. The phone screen displays the time as 9:38, the date as Monday May 4, 2016, and a notification from "QorosSalesAppDev" with the message "测试ios推送1". Below the phone icon is a horizontal dashed line with the text "推送对象" in the center. Underneath this line, there is a section titled "目标平台 (必选)" with four icons: "iOS开发环境" (blue Apple), "iOS生产环境" (black Apple), "Android" (Android robot), and "WinPhone" (Windows logo). Further down is a section titled "目标人群" with five tabs: "广播 (所有人)" (selected, highlighted in blue), "设备标签 (Tag)", "设备别名 (Alias)" (selected, highlighted in blue), "Registration ID", and "用户分群推送". Below these tabs is a form field with two radio buttons: "手动输入" (selected) and "文件上传", followed by an input field containing the text "10000027" and a clear button "x". A placeholder text "添加别名,并按enter键添加" is also visible.

推送后，可以方便的看到推送的结果：

The screenshot shows the 'Push History' section of the Qoros Sales Application. It displays a table of push notifications with columns for '发送时间' (Send Time), '内容' (Content), '类型' (Type), and 'iOS' (2|2), 'Android' (0|0), 'Winphone' (0|0) success counts. Below the table is a summary box for iOS: '2 目标 成功 点击 0'. The detailed log below the summary includes fields like '发送编号' (Delivery Number), 'Message ID', '推送方式' (Delivery Method), '接受对象' (Recipient Object), '推送内容' (Delivery Content), '推送平台' (Delivery Platform), 'Extras', '应用角标(badge)', '通知声音(sound)', and '推送结果' (Delivery Result). The delivery result shows '已发送' (Delivered).

发送时间	内容	类型	iOS	Android	Winphone	操作
2018-07-25 16:49	test push to produ...	别名	2   2	0   0	0   0	<i>i</i> <i>删除</i> <i>编辑</i>

2 | 2 | 0

目标 成功 点击

发送编号	5   5
Message ID	3   4
推送方式	立即推送
接受对象	别名:1
推送内容	test push to production ios
推送平台	ios-product
Extras	{"ios":{"sound":"default","badge":1}}
应用角标(badge)	1
通知声音(sound)	default
推送结果	已发送

## iOS证书将要过期会及时提醒

极光推送在证书快要过期前一个月会发邮件通知管理员的，需要你及时去更新证书：

TODO: 1. 整理进来 【记录】 极光推送：AppKey的应用iOS开发证书到期时间

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-06 23:19:08

# 消息推送开发总结

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:08:52

# iOS消息推送

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:09:23

## Android消息推送

### Android的离线推送

不同的品牌手机中的Android系统，有的还针对消息离线推送进行了特殊的定制，比如：

- 华为
- 小米

在系统级别上支持：app没有运行的情况下，也能收到离线消息。

TODO：

1. 把

【调研】iOS android app被杀掉后 如何实施上报地址位置信息 和 离线推送

整理进来。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2018-02-02 16:09:12

# 调试移动端Webview/H5页面

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 17:22:04

# 调试iOS的Webview

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 17:22:30

# 调试Android的Webview

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 17:22:49

# 通话录音

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 16:23:33

# iOS通话录音

TODO:

1.整理进来

【整理】iOS 电话录音 sdk 方案

【已解决】用飞语FYRtcEngineKit去实现基本的iOS间的语音通话

【已解决】用飞语云平台实现网络电话拨打手机号且带录音

【已解决】使用飞语云平台实现iOS的电话录音

飞语官网没有足够完善的demo，自己在折腾后，写了个demo：

[crifan/feiyuiOSDemo: 飞语云平台iOS点对点通话录音Demo](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 16:44:46

# Android通话录音

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 16:24:01

## APP托管总结

在发布到其他app托管平台期间，会涉及到给对于页面的命名

此处有些实践经验：

TOTO：

1. 把

【整理】项目流程总结

中关于fir.im中的不同版本的命名 整理进来

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2018-02-02 16:48:32

# APP托管平台

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:19:09

**fir.im**

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:19:29

# 蒲公英pgyer

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:19:45

# APP上架总结

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:22:44

# iOS上架

- app上架的地方：
  - 公开版本：苹果官网的唯一的App Store
  - 企业版：无需上架，找个app应用发布的地方供用户下载即可，比如
    - fir.im
    - OTA版：自己提供对应的服务器用于存放ipa和对应的plist
      - 注意必须要支持https才可以
  - AdHoc版本：临时发布供测试
    - 在用Xcode打包之前就要设置好对应的profile，添加iPhone等设备等GUID，发布后，加了GUID的iPhone等设备才能安装和使用
- 需要提供App的测试账号和密码
  - 苹果官方审核人员会登录你的app
  - 测试你的基本功能
    - 看看app内容是否有违规内容
    - 且要确保你的app的稳定性
      - 不会随便就崩溃了
      - 否则影响用户体验，就会被拒

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2018-02-02 21:42:47

# iOS小范围测试版: AdHoc

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:23:25

## iOS上架公开市场：AppStore

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:23:35

# iOS内部发布：企业版

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:24:21

## OTA版

### 坑：打包OTA包时的设置的图片地址也要正确否则无法下载

iOS打包OTA的安装包时，即使只是图片有问题（而ipa没问题），也会导致无法正常下载和安装

详见：[\[已解决\] 企业版的iOS的app去In House打包和OTA发布后有时候无法下载和安装：无法下载应用 此时无法下载 完成 重试](#)

### 坑：打包OTA包时plist中的ipa文件或图片的地址中如果包含特殊字符也会导致无法正常下载

对于ipa或图片的文件地址中如果包含特殊不可见字符的话，也是会导致无法下载的。

详见：

[\[已解决\] 企业版iOS的ipa通过OTA发布后还是无法下载和安装](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2018-02-02 22:09:31

# Android上架

- app上架的地方：自己选择，有很多
  - Android官方的：Google Play
  - 国内的：
    - 应用宝
    - 360
    - 百度
    - 等等
- 无需提供测试账号

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 21:41:19

# APP涉及后期运维总结

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:25:39

## 移动APP数据统计

### 数据统计服务提供商

#### 国外

- flurry
- loclytics
- mixpanel
- Google Analytics for Mobile

#### 国内

- TalkingData
- 友盟统计umeng
- LeanCloud
- 魔方
- Bugly: 能起到部分的收集数据（比如多少用户安装了，每日日活如何等基本的数据的）效果

### 自己搭建数据统计后台

开源自行搭建的选择 == 开源解决方案

- [count.ly](#)

## 总结

单独比较数据统计的服务的效果，目前的看法是：

TalkingData > 友盟 > 腾讯云

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 21:49:50

# TalkingData

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:26:27

# 友盟UMeng

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:27:06

## Bugly

腾讯的崩溃日志收集的Bugly，也可以起到部分的数据统计功能：

- 运营概览
- 用户分析
- 版本分布
- 留存分析
- 使用频率/时长
- 渠道分析
- 渠道分布

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2018-02-02 21:45:20

# 崩溃日志收集

崩溃日志收集 = Crash Log Collection = 崩溃检测

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:30:05

# Bugly

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:28:25

# 应用内问题反馈

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:31:26

# 蒲公英pgyer

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:31:55

# iOS开发总结

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:32:40

# iOS开发语言

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:32:59

# Objective-C

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:56:24

# Swift

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:33:29

# 开发工具

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:35:39

# XCode

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:57:06

# AppCode

[AppCode: Smart Swift/Obj-C IDE for iOS & macOS Development](#)

Smart IDE for iOS/macOS development

[Objective-C 的第三方 IDE AppCode 怎么样? - 知乎](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:57:30

## 苹果开发者账号

做苹果的iOS方面的开发，前提是需要有个 苹果开发者账号，一般也被称为 Apple ID = 苹果开发者ID

这个账号是需要购买的，最便宜的个人版，一年也要99美元的。

有了这个苹果开发者账号后，才可以发布app。

而同时，也会享有一些额外的好处。

## 提前使用新版本

注册了苹果开发者账号后，可以有机会在苹果正式发布Xcode, iOS等最新版本之前，可以去下载试用beta版本

Apple Developer Center特殊待遇：可以先下载一些beta的开发环境

登录

[Apple Developer Center](#)

之后，就可以看到里面有很多在正式发布之前的供开发者内测的beta版的iOS, Xcode等资源：

The screenshot shows the Apple Developer Center's Downloads page. At the top, there's a banner with a warning icon and text: "Important: You must have the latest beta version of Xcode 8 installed on your Mac before using a Restore Image to install iOS or tvOS beta software on a device." Below the banner, a note states: "Your use of Apple beta software is subject to and licensed only under the terms and conditions of the Apple Developer Program License Agreement, including any applicable consent to collect diagnostic data set forth therein. If you have not agreed to the Apple Developer Program License Agreement, you are not permitted to use this software. Refer to the software installation guides for complete instructions." A "Featured Downloads" section highlights "Xcode 8 beta 4", which includes the SDKs for iOS 10 beta, macOS 10.12 beta, watchOS 3 beta, and tvOS 10 beta. The download page shows details: Build 8S188o, Date Aug 1, 2016, and a prominent blue "Download" button.

The screenshot shows the Apple Developer website's download page. The top navigation bar includes links for SUI Mobile, Full Feature..., Framework7..., Documentation..., Demo Apps, SUI Mobile, 拓展组件 - SU..., apple develo..., Download ..., and a search bar. The main content area is titled 'Featured Downloads'.

		Build	Date	
<b>Xcode 8 beta 4</b>	Release Notes	8S188o	Aug 1, 2016	<a href="#">Download</a>
Includes the SDKs for iOS 10 beta, macOS 10.12 beta, watchOS 3 beta, and tvOS 10 beta.				
<b>macOS 10.12 beta 4</b>	Release Notes	16A270f	Aug 1, 2016	<a href="#">Download</a>
Refer to the release notes for complete installation instructions.				
<b>iOS 10 beta 4</b> <span style="border: 1px solid red; padding: 2px;">Preferred</span>	Release Notes	14A532e	Aug 1, 2016	<a href="#">Download</a>
Install Configuration Profile directly on any iOS device and receive OTA updates.				
<b>iOS Restore Images</b>				<a href="#">See all</a>
Install via iTunes.				
<b>watchOS 3 beta 4</b>	Release Notes	14T5302d	Aug 1, 2016	<a href="#">Download</a>
Install configuration profile directly on your device to receive OTA updates.				
<b>tvOS 10 beta 4</b> <span style="border: 1px solid red; padding: 2px;">Preferred</span>	Release Notes	14T5308d	Aug 1, 2016	<a href="#">Download</a>
For Apple TV (4th generation). Apple TV (3rd generation) and earlier are not supported. Install configuration profile directly on your device to receive OTA updates.				
<b>tvOS Restore Image</b>				<a href="#">Download</a>
Install via iTunes.				
<b>macOS Server 5.2 beta 3</b>	Release Notes	16S1124f	Aug 1, 2016	<a href="#">Download</a>

The screenshot shows the Apple Developer website's download page, similar to the previous one but with different content. The top navigation bar and 'Featured Downloads' section are identical.

		Build	Date	
<b>tvOS Restore Image</b>				<a href="#">Download</a>
Install via iTunes.				
<b>macOS Server 5.2 beta 3</b>	Release Notes	16S1124f	Aug 1, 2016	<a href="#">Download</a>
<b>Apple Configurator 2.3 beta 2</b>	Release Notes	3D34	Jul 18, 2016	<a href="#">Download</a>
Mac app for deploying iOS devices and Apple TV in your school or business.				
<b>MobileDevice Installer</b> <span style="border: 1px solid red; padding: 2px;">Required</span>				<a href="#">Download</a>
Install the latest MobileDevice installer before using Apple Configurator 2.3 beta.				
<b>Beta Software</b>				
<b>Safari 10 beta 4</b>	Release Notes	602.143	Aug 1, 2016	<a href="#">See all</a>
Safari 10 beta for OS X El Capitan and OS X Yosemite.				
<b>Parallax Exporter</b>				<a href="#">See all</a>
Use the Parallax Exporter plug-in to preview a layered image while working in Adobe Photoshop.				
<b>Parallax Previewer</b>		1A68	Jan 25, 2016	<a href="#">Download</a>
Use Parallax Previewer to preview layered Photoshop files, assemble individual image layers from PNG files and preview the parallax effect, or preview layered images exported by the Parallax Exporter plug-in.				

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 22:05:33

# 好用的库和工具

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:39:05

# Android开发总结

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:43:50

# 开发语言

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:44:04

# Java

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:44:21

# Kotlin

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:44:29

# 开发工具

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:44:53

# Eclipse+ADT

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:45:14

# Android Studio

Android Studio 简称 AS

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:58:33

# 好用的库和工具

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 15:45:52

## 附录

下面列出相关参考资料。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-02 14:51:49

# 参考资料

- [调研] iOS和Android的移动端应用数据统计
- Mobile Web Development - Web developer guides | MDN
- Web App 和 Native App, 哪个是趋势? - 赵瀚卿的回答 - 知乎
- [已解决] 企业版的iOS的app去In House打包和OTA发布后有时候无法下载和安装: 无法下载应用 此时无法下载完成 重试
- [已解决] 企业版iOS的ipa通过OTA发布后还是无法下载和安装
- [基本解决] iOS页面切图的尺寸距离, 图片, 字体的大小和代码中所使用的iOS设备真实的尺寸的关系
- 【整理】利用极光推送JPush实现消息推送
- [记录] 购买JPUSH极光推送收费版 – 在路上
- 【已解决】iOS的JPUSH生产环境无法推送 – 在路上
- [记录] 到极光推送中设置iOS的Development推送 – 在路上
- [记录] 极光推送JPUSH更换生产证书和Bundle ID – 在路上
- [记录] 手动测试JPUSH极光去企业账号生产环境的app中的消息推送

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2018-02-06 23:08:30