

## 前言

本篇博客从开发的角度来介绍如何开发一个Android App，需要说明一点是，这里只是提供一个如何开发一个app的思路，并不会介绍很多技术上的细节，从整个大局去把握如何去构思一个app的开发，让你对独立开发一款app的时候有个理解，如果有说的不对的地方，欢迎大家拍砖留言讨论。

## 开发环境

Android应用层使用的开发语言是Java，自然需要用到Java运行环境，无论你在Window是系统还是Mac系统都需要安装JDK，并且配置它的环境变量，不懂什么叫环境变量的或者不知道怎么配的，请利用好自己的百度技能自行学习。



## 开发工具

开发工具，Android因为历史原因，前面很长一段时间使用的是Eclipse，我们要开发Android应用的时候需要另外安装Google为它开发的插件ADT，Eclipse这个开源软件很优秀，也很专业，但仅仅只是使用它来开发Android就显得有点鸡肋了，如果没有对Eclipse进行优化的话，会很卡很卡，后面Google实在不愿意寄人篱下，就专门为我们开发者开发了目前很受广大开发者推崇的Android Studio，现在2.0预览版也出来啦，大伙赶紧去下吧。



## 模拟器

Android自带的模拟器一直广受诟病，实在是太卡太慢了，对比人家IOS模拟器是那么的流畅，Android开发者直喊好苦逼啊，不过还好后面出了第三方比原生流畅n倍的模拟器Genymotion，大家可以去下一个个人版的够你平时开发测试用了。最好的办法还是使用真机测试，毕竟真机才是真实的环境。



## Android核心基础

前面是准备工作，想开发一款Android app，你最起码要掌握Android的核心基础知识，针对初学Android的童鞋，想一口吃掉一个胖子是不可能的，还是得把基础打牢之后才能完成独立的开发，Android入门不难，学完基础你只是具备开发app的基本条件，想做一个优秀的app还有很长一段路要走，经验是靠不断实践堆出来的，找一套系统的课程去认真学习一下，在线教育的资源很多，比如慕课网，极客学院都有很多不错的学习资源，童鞋自己择优来学习即可。

推广：<http://edu.csdn.net/course/detail/545>（笔者的课程）

## 产品开发流程

正常的互联网开发app的流程大致如下：

- 产品规划，定产品方向
- 需求调研，产出需求文档
- 需求评审，修订需求文档
- 产品狗画app线框图提供给射鸡师
- 射鸡师根据线框图设计视觉稿
- 程序猿根据视觉稿搭建UI框架
- 程序猿根据需求文档开发功能
- 测试媛编写测试用例，根据排期进行测试
- 程序猿修复回归测试反馈的bug，提交beta版
- 测试通过，提交给运营喵发布到渠道上线

上面是笔者的经验总结，可能不太准确的地方，但大致整个流程是这样，我们开发者要清楚自己在整个产品中充当的角色，明白自己的工作职责即可。

## 快速搭建项目

Android比没有想象那么难，只是刚开始要学习的东西多，一下子消化不了才会比较茫然，笔者写这篇博客就是想帮助大家整理一下开发思路。

快速搭建项目也算是一项技能，而这项技能只能由你自己来完善，刚开始做开发的时候一定会有很多重复性的工作，如果没有意识去提高自己的开发效率的话，可能你的十年工作经验代表的只是一年的经验用了十年而已。

那要怎么做，笔者提供一个自己总结的，仅供参考：

- 定开发规范
- 搭建UI框架
- 选用开发库集成（或自造轮子）
- 第三方服务集成（视情况而定）

### 定开发规范

一个项目一般不会只有一个人独立开发，就算是独立开发，我们也应该定一下开发规范，一般会有什么样的规范？

- 命名规范
- 代码规范
- 代码管理规范

命名规范

命名规范包括：

- 项目命名
- 包命名
- 类命名、成员变量命名
- 资源文件命名

我们做每一个产品都会有相应的命名，不要用中文来命名，最好用能表达中文意思的英文来表示，例如CSDN客户端，可以用CSDNClient作为命名，我们创建项目的时候可以以它作为Application name。

可以看看以前写的这篇文章：

[http://blog.csdn.net/wwj\\_748/article/details/42347283](http://blog.csdn.net/wwj_748/article/details/42347283)

代码规范

代码规范视语言而定，开发android使用的是Java，自然要按照Java的编码规范进行编码，比如命名采用驼峰命名法，编码的时候也要符合Android的开发规范，比如UI线程不做任何耗时的操作，像网络请求、数据库操作都需要放到子线程中去做，只有UI的刷新才在UI线程中做，像这些规范，可以根据自身在项目遇到的问题来定，定规范的好处就是减少踩坑的几率，提高开发的质量。

代码管理

对于一个经常更新迭代的产品，不可能由头到尾不变，这个时候我们需要对代码进行分支管理，最好使用git代码仓库对代码进行管理，作为一个合格的开发者连git都不用实在说不过去，还用svn的童鞋赶紧放弃它投入git的怀抱，它的好处是不言而喻的，自己去体会。

### 搭建UI框架

搭建UI框架需要我们根据产品的导航模式来设计，市场上常用的导航模式有如下图几种：



我们的app如果不出意外一定是其中的一种导航模式，一般线框图出来我们就应该知道即将要开发的app长什么样子，开发者不必等视觉稿和素材出来才开始动工，我们先大致搭个架子，等视觉稿出来之后我们再做调整。

## 选用开发库

一般我们app涉及到的库会有：

- UI框架（比如下拉刷新PullToRefresh、侧滑菜单Slidingmenu）
- 网络请求库（比如okhttp、AndroidAsyncHttp、Volley）
- 数据操作库（比如GreenDao、Ormlite）
- 图片缓存框架（比如Universal-ImageLoader）
- 数据解析库（比如Gson）

之所以要选用这些库，肯定是为了避免重复造轮子，在开发效率的角度来说，选用优秀的开源库能大大缩短开发周期和提高开发效率，但从个人提升角度来看的话，我们可能就成了一个只会用API的程序猿了，如果想提升的话，造轮子或者分析这些优秀的源代码是一个不错的途径。

## 第三方服务集成

我们开发app的时候，肯定会遇到一些需求，比如推送的需求、自动升级、数据统计、社会化分享、用户反馈等等，然而对于一个刚起步的企业或者个人开发者的话，全都要自己去开发的话，那岂不是累死，像推送这种有一定的技术门槛，能做好都能成立一家公司了，所以选用一些第三方服务是一个可选之举。如果说你以后做大了，用第三方怕不好控制，那就自己做呗，有钱任性招兵买马就自己做，谁叫咱有钱呢。

前面这些东西开发一个app够了，开发出来能不能用还得有靠谱的测试，有没有crash，操作流不流畅，体验好不好才会有用户去用。这里不从产品的角度去评判一个app的好坏，程序员要考虑的是从代码层面、性能层面去让我们的app变得更好。

## 云测

我们开发完毕之后，需要给测试工程师进行基本的功能需求测试，他们传统的做法就是根据事先写好的测试用例来做回归测试，再把测试出来的bug反馈给工程师，工程师再去修bug，但这样实在是太不靠谱了，有时候我们太在意功能而忽略了一些更重要的东西，那就是体验，给用户最直接的感受就是你这个app够不够惊艳，够不够流畅，用户可能根本就不在乎你这个功能做的有多牛逼。所以我们更应该从非功能性方向去做测试，我们的目的是让用户用的爽，而不是加一些乱七八糟的功能。那怎么测非功能性的一些因素，这里就要提到『云测』这个东西，因为现在设备太多了，如果公司要买一堆设备来做测试，那得多少成本，况且设备更新得太快，你根本就跟不上，所以就有了云测这个东西，它是一个云测试平台服务，提供了一大批主流机型，我们就直接省去购买设备的成本，还能得到完善的测试报告。

再来说一下它的好处：

- 终端云，省去测试设备购买租赁成本
- 高效率 节省测试人员成本及时间
- 包含兼容性测试、性能测试、功能测试
- 操作简单、详细测试报告生成

这么多好处，你在缺少测试工程师的时候，不去尝试那实在说不过去。

## 打包上线

前面的开发环节、测试环节都没问题之后，你离实现一个app的完整开发就不远了，正常的互联网公司，会把签名打包的apk给到运营，交给他们去写文案，上传到应用渠道，让渠道给我们去首发和推广。如果是个人开发者，这些工作就得我们自己做了。

## 总结

本篇博客从整个app开发流程上去给大家梳理了一遍，没有讲太多技术上的东西，但理解app流程对自己把握产品很有益处，虽然我们只是一个小小的开发者，但如果你有追求，哪天轮到你去负责一个产品的时候，你就应该对开发的每一个环节都了如指掌，因为出了问题，你也知道怎么针对性的去解决。笔者虽然只是一个小小的开发者，但也乐于思考，很多时候不太愿意被别人牵着鼻子走，因为我们的人生就应该把握在自己手里。

如果你觉得文章不错，那就扫码关注一下笔者的公众号吧，期待有志之士的你。

