

移动App

什么是混合移动App开发【重点】

1. 苹果上的软件是如何开发出来的：使用的是 OC、或者使用Swift这门语言
2. 安卓平台上的软件又是如何开发出来的：使用安卓相关的语言开发的，Java，安卓的控件进行开发
3. 苹果和安卓平台上共有的软件是如何开发出来的：腾讯招两套开发人员【开发组】，手机京东
4. 前端移动 App (Application) 开发技术，去开发手机端的应用程序；
5. 前端的混合移动App开发技术，并没有使用 苹果 或 安卓 官方推荐的 开发平台和开发方式，而是抛弃了 官方提供的方式，使用 前端的独有的技术进行移动App开发体验；

什么是移动App开发：通俗的理解，就是把开发Web网站的技术（HTML+CSS+JS），通过某种方式，移植到移动App开发上进行使用，这种利用Web开发技术进行移动端开发体验的方式，叫做混合移动App开发！

关于移动App开发，需要知道的几个概念：

- 原生开发：它的英文单词是（NativeApp），指的就是使用 IOS、Android 官方提供的工具、开发平台、配套语言进行 手机App开发的方式；
- 混合开发：（HybirdApp）就是使用前端已有的技术，HTML + CSS + JS，然后再搭配一些相关的打包编译技术，就能够开发出一个手机App，安装到手机中进行使用；
- 什么是App：App是（Application的缩写），意思是：可安装的应用程序；
- App的分类：
 - 按照平台来划分：
 - PC端：浏览器、代码编辑器、PC端的游戏、听歌的、看视频的、聊天的
 - 移动端：手机QQ、手机微信、手机爱奇艺、亡者农药
 - 按照功能来划分：
 - 游戏：愤怒的小鸡仔、植物大战僵尸、亡者农药.....LOL
 - 应用：非游戏类的软件，支付宝、陌陌、美团外卖、
- App和Web的区别：
 - APP概念：App是（Application的缩写），意思是：可安装的应用程序；
 - 优点：流畅、稳定、基本上一些App都可以脱网运行，用户体验好；
 - 缺点：不能跨平台
 - Web概念：特指那些基于浏览器的web网站（本质：就是网页）
 - 优点：可以跨平台（浏览器天生就是跨平台的）
 - 缺点：没有App流畅、不稳定，受限于网速和网络

为什么要学混合App开发

从程序员的角度分析：

1. 挣钱多（别人不会的你会，别人会的，你精通）
 2. 对于找工作来说：（React Native）市场需求量大，好找工作，提高我们的行业竞争力
 3. 能接触到前端流行的技术和框架（各大公司基本都再用React），注意：再React中我们全部都使用ES6语法（class）
- 前端是一个永恒的行业???（只要世界上还有浏览器的存在，必然需要前端，只不过，随着时间的推移，技术更新换代，可能我们对新技术的要求会越来越高）

- 屌丝的崛起之路：只能做页面 -> Ajax前后台数据交互 -> JQuery、Bootstrap -> webApp -> 三大框架 -> 可以做手机混合App/桌面应用 -> 可以做手机原生App -> 将来或许可以发射火箭发射卫星发射导弹 -> 终极目标：统一全宇宙
- 4. (搞前端App开发) 能购置一批牛逼的设备【苹果笔记本、IOS测试机、安卓手机（三星的、华为、小米）】

从企业的角度分析:(选择合适自身的移动App开发方式)【重点】

- 节省开发成本
- 从工资上：尽最大的可能，压榨员工的剩余劳动力
- 从时间上：因为原生的安卓和IOS开发，它们的开发效率并不是很高，因为原生的代码复杂度比较高，因此原生的开发周期比较慢；如果采用移动App开发，那么，我们的开发周期会很短；因为HTML + CSS + JS 足够简单；（对于前端开发APP来说，有两种方式，其中，比较早的一种，也是比较简单的一种，就是先开发出一个网站，然后再把网站运行一行打包的命令，就能得到一个APP了）

1. 市面上常见的App开发方式

- WebApp：基于浏览器实现的，有特定功能的网站，称作WebApp

- 例如：百度脑图、<https://m.jd.com/>、<https://m.taobao.com/#index>
- 优点：跨平台
- 缺点：依赖网络，有白屏效果，相对来说，用户体验差；不能调用硬件底层得设备，比如摄像头；

- NativeApp：用android和Object-C等原生语言开发的应用

- 优点：体验好；用户使用起来很流畅；非常适合做游戏【性能高】；可以直接调用硬件底层的API；
- 缺点：不能跨平台

- HybirdApp：利用前端所学的知识去开发移动端App，兼具2者的优势

- 优点：能够跨平台；体验会好一些；也能够调用硬件底层的API
- 缺点：相对于原生体验稍微弱一丢丢；不适合做游戏；适合做非游戏类型的手机App；
- 应用场景：

- 注意：使用 Java 或者 IOS 写出来的代码和程序，在最终运行的时候，普通的文本代码，都会被编译为原生的机器码去运行，并不像JS这样，解析执行，Java代码是编译执行的；

2. 三种开发方式的原理和对比



Web App、Hybrid App、Native APP对比

	Web App（网页应用）	Hybrid App（混合应用）	Native App（原生应用）
开发成本	低	中	高
维护更新	简单	简单	复杂
体验	差	中	优
Store或market认可	不认可	认可	认可
安装	不需要	需要	需要
跨平台	优	优	差

3. [谁在使用React Native???](#)

企业如何选择合适的App开发方式

1. 如果这个企业中，曾经使用原生技术开发过一些APP，那么在维护的时候，必然需要使用原生技术来维护
2. 如果企业中，需要做一些游戏级别的应用，那么推荐使用原生，因为原生运行效率高，对耗电量处理的很好；
3. 如果企业做一些应用级别的非游戏软件，比如 淘宝、京东、美团，就可以使用 混合APP了；
4. 在企业中，最主要的是好的点子，如果有了一个好的项目立案，那么最好要立即把这个项目做出来；这时候，使用混合App非常合适，因为开发周期很短，能快速上线，抢先占领市场；【裤衩开发】

企业中项目开发流程

- 需求调研：产品定位、受众群体、市场需求、开发价值；【产出物：需求文档】
- 产品设计：功能模块、流程逻辑；【产出物：设计文档，交互稿】，确定项目的基本功能；
- 项目开发：项目架构、美工、前端、后台、测试【产品的把控】**要理解前后端分离的概念**
- 运营维护：上线试运行、调Bug、微调功能模块、产品迭代

根据需求搞设计，根据设计做开发

企业技术选型 - 几大主流技术之间的关系

1. Angular.js 和 Ionic

- [Angular1官网](#)
- [Angular2官网](#)
- [Ionic 中文网](#)
- [Ionic 英文官网](#)

2. Vue.js 和 Weex

- [Vue.js官网](#)
- [Weex文档](#)
- [Weex - github地址 - 新](#)
- [Weex - github地址 - 旧](#)

3. React.js 和 React-Native

- [React.js英文官网](#)
- [ReactNative中文网](#)
- [ReactNative英文网](#)

Angular, Vue, React 这三个都是前端框架，我们在进行混合App开发的时候，只是用到了这三个框架的【基础语法】而已；

Ionic, Weex, ReactNative 这三个都是打包工具（提供了相关的命令行，只要运行指定的命令，就能够把项目打包成一个手机App出来），能够把我们开发出来的应用，最终打包成一个可安装的手机端程序安装包；同时，这三个东西，也提供了好用的一些小组件，方便我们去构建移动App的用户界面；

前端混合App开发框架

1. Html5+、ReactNative、Weex、Ionic

2. [认识HTML5+](#)

- h5+是一个产业联盟，它有一些互联网成员，专门在中国推广H5

3. [HBuilder官网](#)

开发框架之间的区别

1. Html5+ 和 Ionic

2. ReactNative 和 Weex

使用HBuilder生成安卓应用（在线）

[API地址](#)

Hbuilder这个工具，是一个在线打包工具，使用很方便，不需要在本地配置开发环境；直接将做好的网站，通过一些简单的操作，就能在线打包为一个App出来；

- 在项目上右键 -> 发行 -> 发行为原生安装包

好处：本地不用配置开发环境；操作方便，对于程序员来说不关心打包的过程，打包过程对于我们来说是透明的；

缺点：程序员很少能干预打包的过程；源代码被提交到了云端的服务器，存在项目核心代码被泄露的风险；

环境变量的使用

作用：将需要全局使用的工具或者应用程序，配置到Path环境变量中，可以很方便的通过命令行的形式，在任何想要运行这些应用程序的地方，运行它们；

移动App开发环境配置【重点】

安装最新版本的java jdk

1. 修改环境变量，新增 `JAVA_HOME` 的系统环境变量，值为 `C:\Program Files (x86)\Java\jdk1.8.0_112`，也就是安装JDK的根目录
2. 修改系统环境变量 `Path`，在 `Path` 之后新增 `%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;`
3. 新建系统环境变量 `CLASSPATH`，值为 `.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;`
4. 保存所有的系统环境变量，同时退出系统环境变量配置窗口，然后运行cmd命令行工具，输入 `javac`，如果能出现javac的命令选项，就表示配置成功！

安装Node.js环境

注意：需要安装最新的长期稳定版本，不要实验版本；安装完毕之后的node.js会自动配置到全局系统环境变量中

安装完毕后，可以输入 `node -v` 查看node版本号；

安装C++环境

大多数情况下操作系统自带C++环境，不需要手动安装C++环境；

如果运行报错，则需要手动安装visual studio中的C++环境；

安装Git环境

Git安装完毕后，会自动配置到系统环境变量中；

可以通过运行 `git --version` 来检查是否正确安装和配置了Git的环境变量；

安装Python环境

1. 注意：安装Python时候，只能**安装2.x的版本**，注意勾选安装界面上的 `Add Python to path`，这样才能自动将Python安装到系统环境变量中；
2. 安装完毕之后，可以在命令行中运行 `python`，检查是否成功安装了python。

配置安卓环境

1. 安装 `installer_r24.3.4-windows.exe`，最好手动选择安装到C盘下的android目录
2. 打开安装的目录，将 `android-25`、`android-23` (react-native必须依赖这个)解压后，放到 `platforms` 文件夹下
3. 解压 `platform-tools`，放到 `platform-tools` 文件夹下
4. 【这一步直接忽略即可！】**tools文件夹不解压覆盖也行；解压tools，放到安装根目录中**
5. 解压 `build-tools_r23.0.1-windows.zip` (react-native必须依赖这个)、`build-tools_r23.0.2-windows.zip` (weex必须依赖这个) 和 `build-tools_r23.0.3-windows.zip`，并将解压出来的文件夹，分别改名为版本号 `23.0.1`、`23.0.2` 和 `23.0.3`；在安装目录中新建文件夹 `build-tools`，并将改名为版本号之后的文件夹，放到新创建出来的 `build-tools` 文件夹下
6. 在安装目录中，新建 `extras` 文件夹，在 `extras` 文件夹下新建 `android` 文件夹；解压 `m2respository` 文件夹和 `support` 文件夹，放到新建的 `extras -> android` 文件夹下
7. 配置安装环境变量：在系统环境变量中新建 `ANDROID_HOME`，值为android SDK Manager的安装路径 `C:\Users\liulongbin\AppData\Local\Android\android-sdk`，紧接着，在Path中新增 `;%ANDROID_HOME%\tools;%ANDROID_HOME%\platform-tools;`

RN学习说明

1. ReactNative是基于React这门框架的语法来进行开发的;
2. RN中, 提供了 移动端 专用的一些组件, 这时候, 我们在网页中使用的一些 元素, div, p, img 都不能用了, 只能使用RN固有的组件;
3. 最终, 开发出来的项目, 是要运行到手机上的, 那么, 如何把一个 RN 的项目, 完整的发布到手机上去运行呢, 这里, 需要结合 安卓的 签名打包步骤, 并使用 RN 提供的打包命令, 进行完整 apk 文件的发布; 最终发布出来的就是 Release 版本的项目, 可以上传到应用商店;

配置ReactNative基本开发环境

[搭建基本的开发环境 - 英文官网](#)

[搭建基本的开发环境 - 中文](#)

这两篇文档对比着进行参考, 进行相关的安装;

Node.js开发环境准备

1. 普通安装方式官方网站
2. 配置环境变量

1) node.js的msi包是一路next就可以了

2) 安装完后, 可以在命令行中输入node -v 来查看安装版本和是否安装成功, 再输入npm -v查看npm模块是否正常

3) 配置npm的全局模块 (新建文件夹node_global、node_cache)

npm config set prefix "D:\nodejs\node_global" 》模块配置位置

npm config set cache "D:\nodejs\node_cache" 》缓存文件

4) 配置环境变量

进入环境变量对话框, 在【系统变量】下新建【NODE_PATH】, 输入

D:\nodejs\node_global\node_modules

将【用户变量】下的【Path】修改为 D:\nodejs\node_global

(6) 配置完后, 安装个module测试下, 我们就安装最常用的express模块, 打开cmd窗口输入命令, 进行模块的全局安装

npm install express -g # -g是全局安装的意思

JDK

1.安装NEXT 记住安装路径

2.设置环境变量

1)、新建环境变量JAVA_HOME

变量名: JAVA_HOME

变量值: D:\Program Files\Java\jdk1.8.0_191

变量值是自己的JDK安装目录 (这里的设置跟以前一样)

2) 关键中的关键, 配置环境变量Path

双击Path, 编辑Path环境变量,

点击新建，添加“%JAVA_HOME%\bin”；

再次点击新建，添加“%JAVA_HOME%\jre\bin”。

Android Studio

1. 安装 Android Studio#

首先[下载和安装 Android Studio](#)，国内用户可能无法打开官方链接，请自行使用搜索引擎搜索可用的下载链接。安装界面中选择"Custom"选项，确保选中了以下几项：

- Android SDK
- Android SDK Platform
- Android Virtual Device

然后点击"Next"来安装选中的组件。

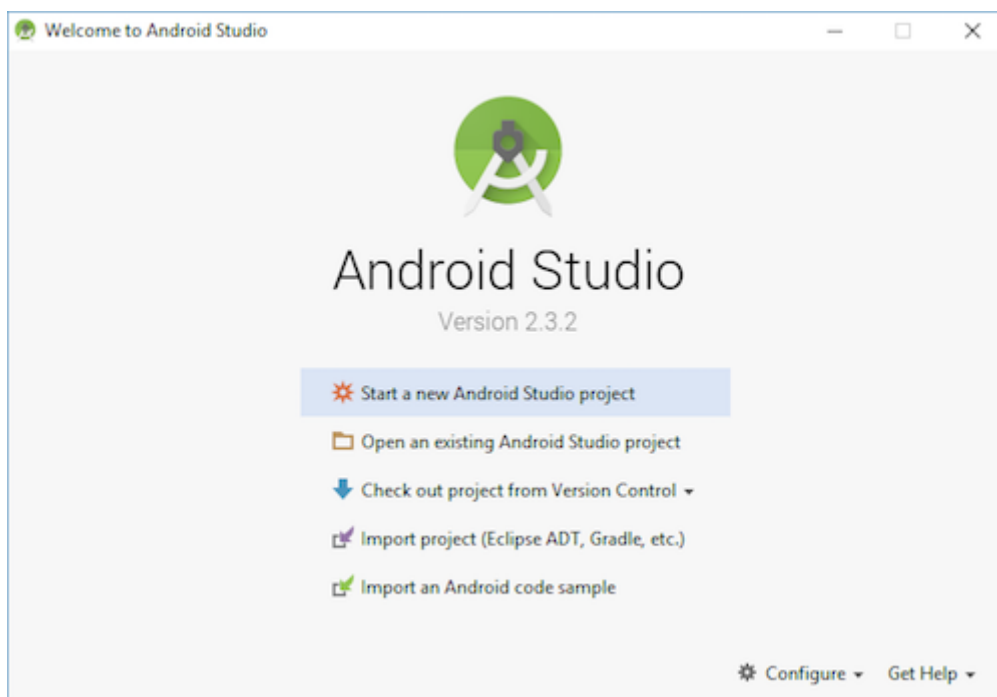
如果选择框是灰的，你也可以先跳过，稍后再来安装这些组件。

安装完成后，看到欢迎界面时，就可以进行下面的操作了。

2. 安装 Android SDK#

Android Studio 默认会安装最新版本的 Android SDK。目前编译 React Native 应用需要的是 Android 10 (Q) 版本的 SDK（注意 SDK 版本不等于终端系统版本，RN 目前支持 android 4.1 以上设备）。你可以在 Android Studio 的 SDK Manager 中选择安装各版本的 SDK。

你可以在 Android Studio 的欢迎界面中找到 SDK Manager。点击"Configure"，然后就能看到"SDK Manager"。



SDK Manager 还可以在 Android Studio 的"Preferences"菜单中找到。具体路径是**Appearance & Behavior** → **System Settings** → **Android SDK**。

在 SDK Manager 中选择"SDK Platforms"选项卡，然后在右下角勾选"Show Package Details"。展开 Android 10 (Q) 选项，确保勾选了下面这些组件（重申你必须使用稳定的代理软件，否则可能都看不到这个界面）：

- Android SDK Platform 29

- `Intel x86 Atom_64 System Image` (官方模拟器镜像文件, 使用非官方模拟器不需要安装此组件)

然后点击"SDK Tools"选项卡, 同样勾选右下角的"Show Package Details". 展开"Android SDK Build-Tools"选项, 确保选中了 React Native 所必须的 29.0.2 版本。你可以同时安装多个其他版本。

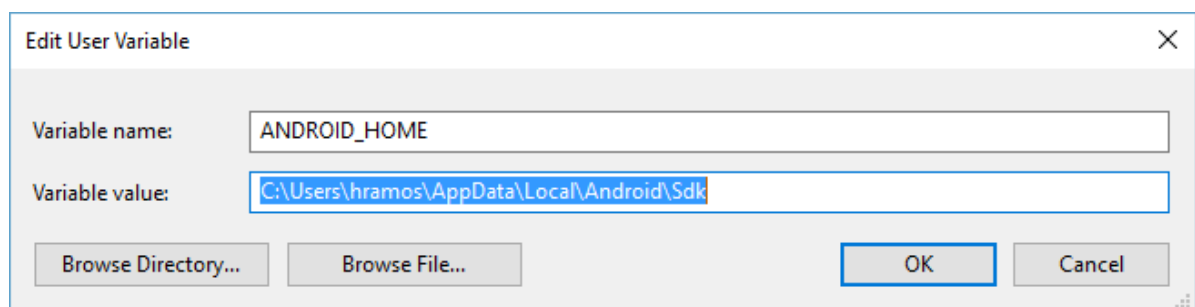
然后还是在"SDK Tools"选项卡, 点击"NDK (Side by side)", 同样勾选右下角的"Show Package Details", 选择 20.1.5948944 版本进行安装。

最后点击"Apply"来下载和安装这些组件。

3. 配置 ANDROID_HOME 环境变量#

React Native 需要通过环境变量来了解你的 Android SDK 装在什么路径, 从而正常进行编译。

打开 控制面板 -> 系统和安全 -> 系统 -> 高级系统设置 -> 高级 -> 环境变量 -> 新建, 创建一个名为 `ANDROID_HOME` 的环境变量 (系统或用户变量均可), 指向你的 Android SDK 所在的目录 (具体的路径可能和下图不一致, 请自行确认):



SDK 默认是安装在下面的目录:

```
C:\Users\你的用户名\AppData\Local\Android\Sdk
```

Copy

你可以在 Android Studio 的"Preferences"菜单中查看 SDK 的真实路径, 具体是 **Appearance & Behavior** → **System Settings** → **Android SDK**。

你需要关闭现有的命令提示窗口然后重新打开, 这样新的环境变量才能生效。

4. 把一些工具目录添加到环境变量 Path#

打开 控制面板 -> 系统和安全 -> 系统 -> 高级系统设置 -> 高级 -> 环境变量, 选中 **Path** 变量, 然后点击 **编辑**。点击 **新建** 然后把这些工具目录路径添加进去: platform-tools、emulator、tools、tools/bin

```
%ANDROID_HOME%\platform-tools%ANDROID_HOME%\emulator%ANDROID_HOME%\tools%ANDROID_HOME%\tools\bin
```

创建新项目#

如果你之前全局安装过旧的 `react-native-cli` 命令行工具, 请使用 `npm uninstall -g react-native-cli` 卸载掉它以避免一些冲突。

使用 React Native 内建的命令行工具来创建一个名为"AwesomeProject"的新项目。这个命令行工具不需要安装, 可以直接用 node 自带的 `npx` 命令来使用:

必须要看的注意事项一：请 不要 在目录、文件名中使用 中文、空格 等特殊符号。请 不要 单独使用常见的关键字作为项目名（如 class, native, new, package 等等）。请 不要 使用与核心模块同名的项目名（如 react, react-native 等）。

必须要看的注意事项二：请 不要 在某些权限敏感的目录例如 System32 目录中 init 项目！会有各种权限限制导致不能运行！

必须要看的注意事项三：请 不要 使用一些移植的终端环境，例如 git bash 或 mingw 等等，这些在windows下可能导致找不到环境变量。请使用系统自带的命令行（CMD或powershell）运行。

```
npx react-native init AwesomeProject
```