# Cordova+Angularjs+Ionic 混合开发

## 一、Cordova+Angularjs+Ionic 混合开发入门讲解

Cordova：

以前叫PhoneGap，是一个可以与手机原生代码交互的框架，主要用于调用手机功能，比如调用拍照功能，定位功能等。

Cordova是贡献给Apache后的开源项目，是从PhoneGap中抽出的核心代码，是驱动PhoneGap的核心引擎。你可以把它们的关系想象成类似于Webkit和Google Chrome的关系

Cordova提供了一组设备相关的API，通过这组API，移动应用能够以JavaScript访问原生的设备功能，如摄像头、麦克风等。

Cordova还提供了一组统一的JavaScript类库，以及为这些类库所用的设备相关的原生后台代码。

Cordova支持如下移动操作系统：iOS, Android,ubuntu phone os, Blackberry, Windows Phone, Palm WebOS, Bada 和 Symbian。

Angularjs：

是一款优秀的前端JS框架，已经被用于Google的多款产品当中。AngularJS有着诸多特性，最为核心的是：MVVM、模块化、自动化双向数据绑定、语义化标签、依赖注入等等

Ionic：

ionic是一个用来开发混合手机应用的，开源的，免费的代码库。可以优化html、css和js的性能，构建高效的应用程序，而且还可以用于构建Sass和AngularJS的优化。ionic会是一个可以信赖的框架。

## 二、Ionic + Angularjs + Cordova（Ionic2环境搭建）

——————-A、安装环境———————

1、下载NodeJs，最好能下载压缩包版的，不要安装版   
2、检查是否安装了Jdk8以上，包含Jdk8，如果没有，请先安装   
3、检查是否下载了Android SDK，如果没有，请先下载   
4、检查是否下载了Gradle-3.4.1，如果没有，请先下载

——————-B、配置环境———————

这里是环境的配置详情，当然这里是我的电脑上的配置，你们看情况配成自己的路径即可。

ANDROID\_HOME=D:/BuildTools/AndroidSDKJAVA\_HOME=C:\Program Files\Java\jdk1.8.0\_131PATH=%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;D:\BuildTools\node-v8.2.1-win-x64;D:/Tools/Core/gradle-3.4.1/bin;%ANDROID\_HOME%/tools;%ANDROID\_HOME%/tools/bin;%ANDROID\_HOME%/platform-tools

——————-C、调试环境———————

1、测试Jdk、Gradle、NodeJs是否配置成功，类似下面的即为成功：

Microsoft Windows [版本 6.1.7601]

版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>java

用法: java [-options] class [args...]

(执行类)

或 java [-options] -jar jarfile [args...]

(执行 jar 文件)

其中选项包括:

-d32 使用 32 位数据模型 (如果可用)

-d64 使用 64 位数据模型 (如果可用)

-server 选择 "server" VM 默认 VM 是 server.

-cp <目录和 zip/jar 文件的类搜索路径>

-classpath <目录和 zip/jar 文件的类搜索路径>

用 ; 分隔的目录, JAR 档案

和 ZIP 档案列表, 用于搜索类文件。

-D<名称>=<值> 设置系统属性

-verbose:[class|gc|jni] 启用详细输出

-version 输出产品版本并退出

-version:<值> 警告: 此功能已过时, 将在

未来发行版中删除。

需要指定的版本才能运行

-showversion 输出产品版本并继续

-jre-restrict-search | -no-jre-restrict-search

警告: 此功能已过时, 将在未来发行版中删除。

在版本搜索中包括/排除用户专用 JRE

-? -help 输出此帮助消息 -X 输出非标准选项的帮助 -ea[:<packagename>...|:<classname>] -enableassertions[:<packagename>...|:<classname>] 按指定的粒度启用断言

-da[:<packagename>...|:<classname>] -disableassertions[:<packagename>...|:<classname>] 禁用具有指定粒度的断言

-esa | -enablesystemassertions 启用系统断言

-dsa | -disablesystemassertions 禁用系统断言

-agentlib:<libname>[=<选项>] 加载本机代理库 <libname>, 例如 -agentlib:hprof

另请参阅 -agentlib:jdwp=help 和 -agentlib:hprof=help

-agentpath:<pathname>[=<选项>] 按完整路径名加载本机代理库

-javaagent:<jarpath>[=<选项>] 加载 Java 编程语言代理, 请参阅 java.lang.instrument

-splash:<imagepath> 使用指定的图像显示启动屏幕

有关详细信息, 请参阅 http://www.oracle.com/technetwork/java/javase/documentation/index.html。

C:\Users\Administrator>javac

用法: javac <options> <source files>

其中, 可能的选项包括:

-g 生成所有调试信息 -g:none 不生成任何调试信息 -g:{lines,vars,source} 只生成某些调试信息 -nowarn 不生成任何警告 -verbose 输出有关编译器正在执行的操作的消息 -deprecation 输出使用已过时的 API 的源位置 -classpath <路径> 指定查找用户类文件和注释处理程序的位置 -cp <路径> 指定查找用户类文件和注释处理程序的位置 -sourcepath <路径> 指定查找输入源文件的位置 -bootclasspath <路径> 覆盖引导类文件的位置 -extdirs <目录> 覆盖所安装扩展的位置 -endorseddirs <目录> 覆盖签名的标准路径的位置 -proc:{none,only} 控制是否执行注释处理和/或编译。 -processor <class1>[,<class2>,<class3>...] 要运行的注释处理程序的名称; 绕过默认的搜索进程

-processorpath <路径> 指定查找注释处理程序的位置 -parameters 生成元数据以用于方法参数的反射 -d <目录> 指定放置生成的类文件的位置 -s <目录> 指定放置生成的源文件的位置 -h <目录> 指定放置生成的本机标头文件的位置 -implicit:{none,class} 指定是否为隐式引用文件生成类文件 -encoding <编码> 指定源文件使用的字符编码 -source <发行版> 提供与指定发行版的源兼容性 -target <发行版> 生成特定 VM 版本的类文件 -profile <配置文件> 请确保使用的 API 在指定的配置文件中可用 -version 版本信息 -help 输出标准选项的提要 -A关键字[=值] 传递给注释处理程序的选项 -X 输出非标准选项的提要 -J<标记> 直接将 <标记> 传递给运行时系统 -Werror 出现警告时终止编译 @<文件名> 从文件读取选项和文件名

C:\Users\Administrator>gradle -v

------------------------------------------------------------Gradle 3.4.1

------------------------------------------------------------

Build time: 2017-03-03 19:45:41 UTC

Revision: 9eb76efdd3d034dc506c719dac2955efb5ff9a93

Groovy: 2.4.7

Ant: Apache Ant(TM) version 1.9.6 compiled on June 29 2015

JVM: 1.8.0\_131 (Oracle Corporation 25.131-b11)

OS: Windows 7 6.1 amd64

C:\Users\Administrator>npm -v

5.3.0

2、安装Ionic、Cordova

1、映射淘宝库

npm install cnpm -g --registry=https://registry.npm.taobao.org

2、安装Cordova到全局库

cnpm install -g cordova3、安装Ionic到全局库

cnpm install -g ionic

3、测试Cordova、Ionic是否安装成功，显示版本号，即可。

C:\Users\Administrator>ionic -v

3.6.0

C:\Users\Administrator>cordova -v

7.0.1

4、创建第一个Ionic2项目（我的NodeJs的版本是“node-v8.2.1”）

1、开始2、运行，输入"cmd"，回车（双引号别带上哈，只是为了区分要输入什么内容）3、输入"c:"，回车4、复制粘贴指令，等待执行完毕

a、ionic start zmkm --type=ionic-angular --skip-deps --skip-link --cordova

指令解释:

zmkm 项目名称

--type表示创建的App的框架类型

--skip-deps表示跳过安装依赖

--skip-link表示不绑定Ionic帐号

--cordova表示集成Cordova5、复制粘贴指令，等待执行完毕（每一行为一行指令，千万别当成一行复制了）

a、cd zmkm

b、cnpm install --save-dev --save-exact @ionic/cli-plugin-ionic-angular@latest, 没有报错说明即将成功

c、ionic serve, 如果出现了下图的网页，说明成功了

http://thumbnail0.baidupcs.com/thumbnail/e60ca00cb67070cb53ac97dffd2bd9ed?fid=1177877023-250528-72786870979589&time=1501491600&rt=pr&sign=FDTAER-DCb740ccc5511e5e8fedcff06b081203-bs9dY%2fHYqhr%2bjOhntZCSTqUeCH8%3d&expires=8h&chkbd=0&chkv=0&dp-logid=4910865478256415026&dp-callid=0&size=c1366\_u768&quality=90&vuk=1177877023&ft=image

6、最终操作

//切换为国内的npm

a、npm install -g nrm

b、nrm ls

c、nrm use taobao

d、cnpm install --save-dev --save-exact @ionic/cli-plugin-cordova@latest

e、cordova platform add android --nofetch, 成功了

f、cordova build android --verbose,调试模式运行,输出日志

f步骤过了以后，有两种情况:

第一种是报错，构建失败(BUILD FAILED)

第二种是缓慢的走完全程，构建成功（BUILD SUCCESSFUL）

7、导入Android Studio或者其他的开发平台IDE，这个就问度娘吧，怎么导项目应该不难